



TESIS KI142502

Komputasi Numerik Model Penyobekan Dinamis Lembaran Kertas

ASTRIS DYAH PERWITA

NRP. 5114201066

PEMBIMBING I

Dr. Darlis Herumurti, S.Kom, M.Kom.

NIP. 197712172003121001

PEMBIMBING II

Rully Soelaiman, S.Kom, M.Kom.

NIP. 197002131994021001

PROGRAM STUDI MAGISTER

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2017

[Halaman ini sengaja dikosongkan]



THESIS KI142502

Numerical Scheme Computation for Paper Sheet Adaptive Tearing

ASTRIS DYAH PERWITA

NRP. 5114201066

SUPERVISOR I

Dr. Darlis Herumurti, S.Kom, M.Kom.

NIP. 197712172003121001

SUPERVISOR II

Rully Soelaiman, S.Kom, M.Kom.

NIP. 197002131994021001

MASTER PROGRAM

DEPARTEMENT OF INFORMATICS

FACULTY OF INFORMATION TECHNOLOGY

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2017

[Halaman ini sengaja dikosongkan]

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

oleh:

ASTRIS DYAH PERWITA

Nrp. 5114201066

Dengan judul :

Komputasi Numerik Model Penyobekan Dinamis Lembaran Kertas

Tanggal Ujian : 5-1-2017

Periode Wisuda : 2016 Gasal

Disetujui oleh:

Dr. Darlis Heru Murti, S.Kom, M.Kom
NIP. 197712172003121001

(Pembimbing 1)

Rully Soelaiman, S.Kom, M.Kom
NIP. 197002191994021001

(Pembimbing 2)

Dr. Eng. Nanik Suciati, S.Kom, M.Kom
NIP. 197104281994122001

(Penguji 1)

Wijayanti Nurul Khotimah, S.Kom, M.Sc
NIP. 198603122012122004

(Penguji 2)

Ridho Rahman Hariadi, S.Kom, M.Sc
NIP. 198701032014041001

(Penguji 3)

an. Direktur Program Pascasarjana
Asisten Direktur

Prof. Dr. Ir. Tri Widjaja, M.Eng.
NIP. 196110211986031001



Direktur Program Pasca Sarjana,

Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D.
NIP. 196012021987011001

[Halaman ini sengaja dikosongkan]

Komputasi Numerik Model Penyobekan Dinamis Lembaran Kertas

Nama Mahasiswa : Astris Dyah Perwita

NRP : 5114201066

Pembimbing : 1. Dr. Darlis Herumurti, S.Kom, M.Kom.

2. Rully Soelaiman, S.Kom, M.Kom.

ABSTRAK

Untuk memproduksi detail dan variasi retakan pada berbagai macam benda nyata masih dirasa sangat sulit. Sementara bila membangun model dengan metode Finite Element tradisional, maka penerapan yang dilakukan juga cukup kompleks karena membutuhkan pembaharuan setiap sebuah retakan atau sobekan terbentuk. Oleh karena itu, pada penelitian ini akan berfokus pada teknik simulasi retakan dan sobekan dengan arah dan besaran yang tepat sesuai dengan sifat atau properti yang dikalkulasi dengan perhitungan matematis pada bidang tipis khususnya kertas sesuai dengan metode dan parameter yang tepat.

Untuk dapat menghasilkan model sobekan yang tepat, maka diperlukan suatu skema numerik yang dapat memenuhi kebutuhan (*requirement*) tersebut. Penelitian ini akan melibatkan perhitungan metode Finite Element modifikasi yang melibatkan elemen-elemen pengayaan pada proses eksekusinya. Proses yang dilakukan akan berlangsung dinamis dengan memperhatikan hasil sobekan setiap pada setiap waktu.

Penelitian ini berfokus pada komputasi numerik model penyobekan dinamis lembaran kertas dengan memperhatikan variabel-variabel yang mempengaruhinya. Metode yang diimplementasikan akan menghasilkan model dengan proses *assembly* matrik kekakuan yang konvergen terhadap fungsi waktu. Hasil uji coba penelitian ini menyatakan bahwa rata-rata waktu komputasi pembentukan sobekan pada material tanpa inklusi adalah 21,17 detik dan lebih cepat dari pada pembentukan sobekan dengan inklusi dengan rata-rata waktu komputasi 30,57 detik. Sementara rata-rata waktu yang dibutuhkan untuk inklusi lingkaran adalah

24,75 detik dan lebih cepat daripada inklusi linier dengan rata-rata waktu komputasi 36,39 detik.

Kata kunci: grafika, matrik kekakuan, propagasi sobekan, modifikasi Finite Element

Numerical Scheme Computation for Paper Sheet Adaptive Tearing

Student's Name : Astris Dyah Perwita
Student's ID : 5114201066
Supervisor : 1. Dr. Darlis Herumurti, S.Kom, M.Kom.
2. Rully Soelaiman, S.Kom, M.Kom.

ABSTRACT

To produce detail and variation of tearing and cracking on various object is challenging. Meanwhile using traditional Finite Element Method will need such a complex computation caused by updated value for every crack made. To overcome such problems, research focused on tearing and cracking simulation technique with to predict precise tear amount and angle depend on the variables value on thin plates such as paper.

To generate tearing model that fulfill the requirement, numeric scheme will be needed. Modification of Finite Element Method along with enrichment element will be adapted in this research. And this method will determine which variables take effect on the results. Every process will be done adaptively depending on the existing results for each iteration.

This research will be focused on numerical computation on paper sheet with variables affects. Experiments generate a model which stiffness matrix convergent for each time function. The stiffness matrix will also convergent for each variation of elastic variables. Experiments show that computation time for tearing model will works faster without inclusion as it works on average 21.17 seconds than with inclusion with average 30.57 seconds. It also shows that computation time for tearing model with circular inclusion works faster than linear inclusion with 24.75 seconds for circular inclusion and 36.39 seconds for linear inclusion on average.

Keyword: graphic, stiffness matrix, tearing propagation, FEM modification

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur kepada Allah SWT atas segala karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan tesis yang berjudul “Komputasi Numerik Model Penyobekan Dinamis Lembaran Kertas”.

Harapan dari penulis semoga apa yang tertulis di dalam buku tesis ini dapat bermanfaat dan dapat memberikan kontribusi yang nyata bagi pengembangan ilmu pengetahuan saat ini. Dalam pelaksanaan penelitian dan penyusunan buku tesis penulis mendapatkan banyak bantuan dan dukungan dari berbagai pihak. Tanpa mengurangi rasa hormat, penulis ingin mengucapkan banyak terima kasih kepada:

1. Bapak Rully Soelaiman S.Kom, M.Kom, dan Bapak Dr. Darlis Herumurti, S.Kom, M.Kom. selaku dosen pembimbing yang telah memberikan banyak arahan, bantuan, nasihat, dan perhatian terhadap penelitian penulis.
2. Bapak Waskito Wibisono S.Kom., M.Eng., Ph.D. dan Ibu Chastine Fatichah selaku Ketua dan Sekretaris Program Studi Pascasarjana Teknik Informatika yang telah memberikan kesempatan belajar di Teknik Informatika di tingkat pascasarjana.
3. Pihak Pascasarjana ITS yang telah memberikan kesempatan belajar dalam bentuk Beasiswa Fresh Graduate di ITS Surabaya.
4. Para penulis artikel ilmiah yang karyanya direferensi oleh tesis ini sehingga membantu mempermudah terlaksananya penelitian dan penulisan tesis.
5. Keluarga dan rekan-rekan yang selalu memberikan dukungan moral, material, dan doa sehingga penulis mendapatkan kelancaran untuk menyelesaikan tesis ini.
6. Serta pihak-pihak lain yang tidak dapat disebutkan satu-per-satu disini yang telah banyak membantuk penulis dalam penyusunan tesis ini.

Penulis telah berusaha sebaik-baiknya dalam melakukan penelitian dan menyusun tesis ini. Namun, penulis mohon maaf atas kekurangan maupun kesalahan yang ada pada tesis ini baik dari proses maupun hasil penelitian. Kritik

dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Desember 2016

Astris Dyah Perwita

DAFTAR ISI

LEMBAR PENGESAHAN.....	iv
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Perumusan masalah	3
1.3 Tujuan dan manfaat penelitian	3
1.4 Kontribusi penelitian	4
1.5 Batasan masalah.....	4
BAB 2 KAJIAN PUSTAKA	5
2.1. Tearing dan cracking	5
2.2. Metode Finite Element.....	6
2.2.1. eXtended Finite Element Method	6
2.2.2. Elemen Pengayaan	8
a. Pengayaan Titik Inklusi.....	8
b. Pengayaan Titik Retak dan Heaviside	9
2.2.3. Metode Level Set.....	11
2.3. Stiffness Matrix	13
2.4. Model Sobekan dengan J-Integral	15
BAB 3 METODOLOGI PENELITIAN	19
3.1. Alur Metode Penelitian	19
3.1.1. Studi Literatur Model Peretakan dan Sobekan.....	19
3.1.2. Studi Literatur Metode Finite Element	20
3.1.3. Perancangan Model Sobekan.....	20
3.1.4. Implementasi Model Sobekan	20
3.1.5. Uji Coba dan Analisis Model Sobekan	20
3.2. Rancangan Sistem	21
3.2.1. Masukan	21
3.2.2. Fungsi Utama	23

3.2.3.	Fungsi connectivity.....	25
3.2.4.	Fungsi LevelSet.....	26
3.2.5.	Fungsi calcDOF.....	30
3.2.6.	Fungsi enrElem.....	31
3.2.7.	Fungsi stiffnessMatrix dan updateStiffness	32
3.2.8.	Fungsi forceVector	37
3.2.9.	Fungsi boundaryCond.....	38
BAB 4 UJI COBA DAN PEMBAHASAN		43
4.1.	Lingkungan Uji Coba.....	43
4.2.	Skenario Uji Coba	43
4.2.1.	Uji Coba Model Sobekan Tanpa Inklusi	44
4.2.2.	Uji Coba Model Sobekan dengan Inklusi	51
4.3.	Analisis Hasil Uji Coba	63
BAB 5 KESIMPULAN DAN SARAN.....		65
5.1.	Kesimpulan	65
5.2.	Saran	65
DAFTAR PUSTAKA		67
LAMPIRAN A.....		69
LAMPIRAN B		71
LAMPIRAN C.....		73
BIODATA PENULIS.....		75

DAFTAR GAMBAR

Gambar 2.1 Node-node elemen pengayaan. Fungsi Heaviside ditunjukkan dengan titik bulat dan titik sobekan ditunjukkan dengan titik kotak (Pais, 2011).....	9
Gambar 3.1 Alur metodologi penelitian.....	19
Gambar 3.2 Diagram Alir Model Sobekan	22
Gambar 3.3. Pseudocode fungsi utama	24
Gambar 3.4 <i>Pseudocode</i> Fungsi connectivity	25
Gambar 3.5 <i>Pseudocode</i> Inisialisasi elemen dengan metode Level Set	26
Gambar 3.6 <i>Pseudocode</i> Inisialisasi elemen dengan metode Level Set (2).....	27
Gambar 3.7 <i>Pseudocode</i> Inisialisasi elemen dengan metode Level Set (3).....	28
Gambar 3.8 <i>Pseudocode</i> Inisialisasi elemen dengan metode Level Set (4).....	29
Gambar 3.9 Pseudocode fungsi untuk mendefinisikan PSI (1).....	29
Gambar 3.10 <i>Pseudocode</i> fungsi untuk mendefinisikan PSI (2)	30
Gambar 3.11. Pseudocode untuk menghitung Degree of Freedom	31
Gambar 3.12.Pseudocode untuk membentuk elemen pengayaan (1)	31
Gambar 3.13 <i>Pseudocode</i> untuk membentuk elemen pengayaan (2)	32
Gambar 3.14 <i>Pseudocode</i> untuk membangun matrik kekakuan (1)	32
Gambar 3.15 <i>Pseudocode</i> untuk membangun matrik kekakuan (2)	33
Gambar 3.16 <i>Pseudocode</i> untuk membangun matrik kekakuan (3)	34
Gambar 3.17 <i>Pseudocode</i> untuk membangun matrik kekakuan (4)	35
Gambar 3.18. <i>Pseudocode</i> untuk membangun matrik kekakuan (5)	36
Gambar 3.19 <i>Pseudocode</i> untuk membangun matrik kekakuan (6)	37
Gambar 3.20. Pseudocode fungsi forceVector(1)	37
Gambar 3.21 <i>Pseudocode</i> fungsi forceVector (2).....	38
Gambar 3.22 <i>Pseudocode</i> untuk mendefinisikan <i>boundary condition</i>	38
Gambar 3.23 <i>Pseudocode</i> fungsi penghitungan faktor intensitas tegangan dengan J-Integral (1).....	39
Gambar 3.24 <i>Pseudocode</i> fungsi penghitungan faktor intensitas tegangan dengan J-Integral (2).....	40
Gambar 3.25 <i>Pseudocode</i> fungsi penghitungan faktor intensitas tegangan dengan J-Integral (3).....	41

Gambar 3.26. <i>Pseudocode</i> fungsi growCrack (1).....	41
Gambar 3.27. <i>Pseudocode</i> fungsi growCrack (2).....	42
Gambar 4.1. Persebaran Stress pada proses sobekan tanpa inklusi variasi 1 (a) iterasi ke-1 (b) iterasi ke-10.....	45
Gambar 4.2. Persebaran Stress pada proses sobekan tanpa inklusi variasi 1 (a) iterasi ke-20 (c) iterasi ke-30	46
Gambar 4.3 Grafik implementasi <i>assembly</i> matrik kekakuan terhadap fungsi waktu percobaan tanpa inklusi variasi 1	46
Gambar 4.4. Persebaran Stress pada proses sobekan tanpa inklusi variasi 2 (a) iterasi ke-1 (b) iterasi ke-10.....	47
Gambar 4.5 Persebaran Stress pada proses sobekan tanpa inklusi variasi 2 (a) iterasi ke-20 (b) iterasi ke-30.....	48
Gambar 4.6. Grafik implementasi <i>assembly</i> matrik kekakuan terhadap fungsi waktu percobaan tanpa inklusi variasi 2.....	48
Gambar 4.7. Persebaran Stress pada proses sobekan tanpa inklusi variasi 3 (a) iterasi ke-1 (b) iterasi ke-10.....	49
Gambar 4.8. Persebaran Stress pada proses sobekan tanpa inklusi variasi 3 (a) iterasi ke-20 (b) iterasi ke-30.....	50
Gambar 4.9. Grafik implementasi <i>assembly</i> matrik kekakuan terhadap fungsi waktu percobaan tanpa inklusi variasi 3.....	50
Gambar 4.10. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 1 (a) iterasi ke-1 (b) iterasi ke-10	51
Gambar 4.11. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 1 (a) iterasi ke-20 (b) iterasi ke-30	52
Gambar 4.12. Grafik implementasi <i>assembly</i> matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi lingkaran variasi 1	52
Gambar 4.13. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 2 (a) iterasi ke-1 (b) iterasi ke-10	53
Gambar 4.14. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 2 (a) iterasi ke-20 (b) iterasi ke-30	54
Gambar 4.15. Grafik implementasi <i>assembly</i> matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi lingkaran variasi 2.....	54

Gambar 4.16. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 3 (a) iterasi ke-1 (b) iterasi ke-10	55
Gambar 4.17. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 3 (a) iterasi ke-20 (b) iterasi ke-30	56
Gambar 4.18. Grafik implementasi <i>assembly</i> matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi lingkaran variasi 3	56
Gambar 4.19. Persebaran Stress pada proses sobekan inklusi linier variasi 1 (a) iterasi ke-1 (b) iterasi ke-10	57
Gambar 4.20. Persebaran Stress pada proses sobekan inklusi linier variasi 1 (a) iterasi ke-20 (b) iterasi ke-30	58
Gambar 4.21. Grafik implementasi <i>assembly</i> matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi linier variasi 1	58
Gambar 4.22. Persebaran Stress pada proses sobekan inklusi linier variasi 2 (a) iterasi ke-1 (b) iterasi ke-10	59
Gambar 4.23. Persebaran Stress pada proses sobekan inklusi linier variasi 2 (a) iterasi ke-20 (b) iterasi ke-30	60
Gambar 4.24. Grafik implementasi <i>assembly</i> matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi linier variasi 2	60
Gambar 4.25. Persebaran Stress pada proses sobekan inklusi linier variasi 3 (a) iterasi ke-1 (b) iterasi ke-10	61
Gambar 4.26. Persebaran Stress pada proses sobekan inklusi linier variasi 3 (a) iterasi ke-20 (b) iterasi ke-30	62
Gambar 4.27. Grafik implementasi <i>assembly</i> matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi linier variasi 3	62
Gambar A.1. Referensi nilai konstan modulus Young	69
Gambar B.1 Konstanta elastisitas berdasarkan Mechanic of Materials	71
Gambar C.1. Ilustrasi model sobekan tanpa inklusi (a) Finite Element Mesh dan elemen pengayaan (b) Mesh deformasi.....	73
Gambar C.2. Ilustrasi model sobekan dengan inklusi lingkaran (a) Finite Element Mesh dan elemen pengayaan (b) Mesh deformasi	73
Gambar C.3. Ilustrasi model sobekan dengan inklusi linier (a) Finite Element Mesh dan elemen pengayaan (b) Mesh deformasi	74

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1. Masukan model sobekan	44
Tabel 4.2. Masukan untuk percobaan sobekan tanpa inklusi variasi 1	45
Tabel 4.3. Masukan untuk percobaan sobekan tanpa inklusi variasi 2	47
Tabel 4.4. Masukan untuk percobaan sobekan tanpa inklusi variasi 3	49
Tabel 4.5. Masukan untuk percobaan sobekan dengan inklusi lingkaran variasi 1	51
Tabel 4.6. Masukan untuk percobaan sobekan dengan inklusi lingkaran variasi 2	53
Tabel 4.7. Masukan untuk percobaan sobekan dengan inklusi lingkaran variasi 3	55
Tabel 4.8. Masukan untuk percobaan sobekan dengan inklusi linier variasi 1	57
Tabel 4.9. Masukan untuk percobaan sobekan dengan inklusi linier variasi 2	59
Tabel 4.10. Masukan untuk percobaan sobekan dengan inklusi linier variasi 3 ...	61
Tabel 4.11. Waktu komputasi uji coba (1)	63
Tabel 4.12. Waktu komputasi uji coba (2)	64

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1 Latar belakang

Simulasi retakan secara dinamis memiliki sejarah dan perkembangan yang panjang pada bidang grafika komputer. Namun memproduksi detail dan variasi retakan pada berbagai macam benda nyata masih dirasa sangat sulit (Pfaff, et al., 2014). Sementara bila membangun model dengan metode Finite Element tradisional, maka penerapan yang dilakukan juga cukup kompleks (Pais, et al., 2010). Oleh karena itu, pada penelitian ini akan berfokus pada teknik simulasi retakan dan sobekan dengan arah dan besaran yang tepat sesuai dengan sifat atau properti yang dikalkulasi dengan perhitungan matematis pada bidang tipis khususnya kertas sesuai dengan metode dan parameter yang tepat.

Pada tahun 1988, Terzopoulos dan Fleischer mempresentasikan teknik pemodelan *viscoelastic* dan *plastic deformation*. Metode ini diterapkan dengan tiga metrik tensor fundamental untuk merepresentasikan fungsi energi yang dapat diukur untuk deformasi lekukan, bagian kulit, maupun volume sebuah obyek. Fungsi energi ini yang memungkinkan deformasi model secara berkelanjutan sehingga dapat disimulasikan dalam beberapa metode diskrit. Salah satu dari bentuk metode ini diaplikasikan dengan memperhatikan teknik pembeda terbatas yang dikontrol oleh *continuity spline*. Formulasi ini memungkinkan pembentukan retakan dalam wilayah terbatas sehingga retakan ini dapat dianimasikan dengan koefisien elastis antara titik-titik yang berdekatan sehingga menghasilkan retakan saat jaraknya mencapai batas ambang. (O'Brien, 2000).

Saat sebuah obyek dikenai tekanan, bidang tipis dengan jenis material yang kaku seperti kertas dan lempeng besi akan cenderung menggumpal, membentuk pola yang saling berhubungan dengan lipatan-lipatan dan bentuk-bentuk kerucut yang ganjil. Pola ini terjadi akibat adanya interaksi antara tingkat ketahanan material yang rendah terhadap pembengkokan dan tingkat kekakuan material yang tinggi. (Narain, et al., 2013).

Pada penelitian yang dilakukan Busyarev(2013), diajukan kerangka kerja yang ditujukan untuk membangun visualisasi perobekan terhadap bidang tipis yang

terdiri atas lebih dari satu lapisan. Model dibangun atas empat proses yaitu pembangunan jejaring dasar, pembangunan model retakan, penerapan metode penghalusan, dan pemisahan *layer* untuk material berlapis. Namun penelitian ini tidak dilakukan pendekatan mendetail terhadap parameter-parameter pembentuknya. Selain itu penelitian cenderung berfokus terhadap tujuan utamanya yaitu membentuk model untuk bidang tipis yang berlapis. Sehingga evaluasi yang dibahas adalah perbandingan hasil antara bidang berlapis dan bidang yang tidak berlapis.

Finite Element harus sesuai dengan *mesh* yang direpresentasikan, oleh karena itu *mesh* di sekitar ujung retakan harus dibentuk ulang saat sebuah retakan baru terbentuk. Pada beberapa kasus, pergerakan yang diakibatkan diskontinuitas yang dihitung dengan metode Finite Element tidak praktis untuk diterapkan akibat perlunya pembaharuan untuk setiap perubahan pada geometri dari diskontinuitas yang terjadi. Oleh karena itu beberapa penelitian menerapkan teknik penerapan Finite Element untuk memodelkan retakan tanpa adanya *remeshing* untuk meminimalkan kerja komputasi (Moes, et al., 1999).

Teknik tanpa *remeshing* ini diterapkan dengan memanfaatkan partisi dari sifat *unity* dari Finite Element. Konsep ini memungkinkan adanya fungsi pengayaan secara lokal yang dapat dengan mudah digabungkan ke dalam pendekatan Finite Element. Pendekatan ini dilakukan sedemikian rupa sehingga *degree of freedom* dari elemen pengayaan ini terhubung dengan *mesh*. Untuk kasus analisis retakan atau sobekan, fungsi pengayaan akan difokuskan pada bidang yang berdekatan dengan titik retakan dan fungsi diskontinuitas untuk merepresentasikan perubahan di sepanjang retakan (Melenk & Babuska, 1996).

Untuk memodelkan diskontinuitas kuat yang merepresentasikan bidang retakan dibutuhkan dua jenis titik pengayaan. Pertama adalah fungsi Heaviside yang merepresentasikan diskontinuitas yang mengalir menjauhi titik ujung retakan. Yang kedua adalah perubahan ujung retakan yang bersifat elastis linier yang digunakan untuk memperhitungkan diskontinuitas ujung retakan. Retakan direpresentasikan pada *mesh* secara independen oleh elemen pengayaan sehingga geometri dari retakan dapat diperbaharui tanpa perlu menambah atau membentuk ulang *mesh* baru pada domain. Untuk kasus antarmuka material, elemen pengayaan

diterapkan dengan memperhatikan gabungan jarak yang dihasilkan dari diskontinuitas rendah dan nilai absolut dari nilai fungsi (Pais, et al., 2010).

Untuk dapat menghasilkan tahapan komputasi yang akurat dan efisien, maka diperlukan suatu skema numerik yang dapat memenuhi kebutuhan atau *requirement* yang sesuai.

Berdasarkan penelitian-penelitian sebelumnya, maka diajukan kerangka kerja penyobekan dengan nilai-nilai parameter yang sesuai sehingga terbentuk model yang relatif dan efisien terhadap proses perubahan yang terjadi. Setiap perubahan yang terjadi pada objek merupakan proses yang dikerjakan berdasarkan hasil dari proses sebelumnya. Hal ini dilakukan untuk menghasilkan model yang dinamis. Diharapkan dengan adanya pendekatan terhadap nilai-nilai parameter yang optimal dan adanya elemen pengayaan yang menghemat komputasi, akan didapatkan model yang lebih efisien dibandingkan dengan penelitian-penelitian yang telah dilakukan sebelumnya.

1.2 Perumusan masalah

Permasalahan penyusunan skema komputasi numerik model penyobekan dinamis lembaran kertas meliputi hal-hal sebagai berikut:

1. Bagaimana representasi obyek lembaran kertas menjadi domain komputatif yang dapat mengalami perubahan secara dinamis?
2. Bagaimana bentuk penerapan perhitungan matematis dengan memanfaatkan elemen pengayaan pada domain sehingga dapat membentuk sobekan dan perubahan posisi domain awal?
3. Bagaimana pengaruh variabel-variabel yang ada pada model terhadap hasil sobekan yang terbentuk?

1.3 Tujuan dan manfaat penelitian

Tujuan dari penelitian ini adalah menghasilkan komputasi numerik model penyobekan lembaran kertas yang berlaku secara dinamis sehingga didapatkan skema yang tepat, relatif, dan efisien.

1.4 Kontribusi penelitian

Penelitian ini memiliki kontribusi membangun kerangka kerja model penyobekan dengan memanfaatkan elemen pengayaan pada metode modifikasi Finite Element.

1.5 Batasan masalah

Permasalahan yang dibahas pada penelitian ini memiliki beberapa batasan, yaitu:

1. Sobekan dialirkan pada arah koordinat x,
2. Variasi variabel yang diterapkan adalah variasi yang berkenaan dengan modulus Young dan rasio Poisson sebagai variabel yang menentukan elastisitas sebuah objek,
3. Objek kertas yang digunakan merupakan representasi objek rectangular 2D,
4. Jumlah iterasi pada uji coba adalah 30 kali.
5. Program XFEM yang digunakan adalah adaptasi dari program `xfemQuasiStatic` dari Matthew Jon Pais yang dapat diakses pada <http://sites.google.com/site/matthewjpais/Home>.

BAB 2

KAJIAN PUSTAKA

Bab ini berisi pembahasan tentang kajian dan dasar teori yang digunakan sebagai teori pendukung penelitian. Adapun pembahasannya meliputi *tearing* dan *cracking*, metode Finite Element, dan model sobekan.

2.1. Tearing dan cracking

Retakan atau *fracture* telah dipelajari secara luas baik dari segi mekanisnya maupun teknik-teknik pembangunan, simulasi, dan analisa *behavior* dari banyak materi. Telah banyak teori yang menjelaskan pembentukan sebuah titik retakan dan persebarannya. Teori-teori ini telah menerapkan berbagai metode numerik termasuk elemen terbatas (*finite element*) dan *finite difference methods*, persamaan integral batas, dan simulasi partikel molekular. Aplikasi rekayasa pembentukan model grafika membutuhkan simulasi yang menampilkan dan memprediksi *behavior* dari kejadian nyata dalam bentuk yang akurat dan dapat diandalkan. Pada grafika komputer maupun animasi komputer, hal yang terpenting adalah pengimplementasian bentuk retakan pada sebuah bidang (O'Brien & Hodgins, 1999).

Simulasi sobekan pada domain grafis memiliki banyak kompleksitas dalam pemodelannya. Setiap obyek memiliki tingkah laku atau *behavior* yang unik dalam menampilkan efek dari sobekan pada bidangnya. *Behavior* pada obyek-obyek ini dapat dimodelkan dengan menggunakan *thin-shell model* yang merupakan sebuah representasi obyek dalam bidang datar. *Thin-shell model* dapat direpresentasikan dengan jejaring atau *mesh* segitiga dalam proses pemodelan grafis (Pfaff, et al., 2014).

Program yang bertujuan untuk memodelkan sobekan dengan *framework* Finite Element tradisional adalah sebuah tantangan tersendiri. Pada dasarnya, *framework* Finite Element dimodifikasi untuk dapat bekerja pada diskontinuitas yang terjadi akibat adanya sobekan, inklusi, maupun elemen kosong atau *void*. Dibutuhkan penyesuaian terhadap *framework* Finite Element agar dapat bekerja pada domain *mesh* yang merepresentasikan objek. Pada setiap iterasi dari aliran

sobekan, setidaknya domain pada sekitar titik retak harus disesuaikan agar sobekan yang baru dapat direpresentasikan dengan baik secara geometris (Pais, et al., 2010).

2.2. Metode Finite Element

Sebuah model simulasi sebuah obyek biasanya dibangun atas model kontinyu. Namun pada kenyataannya, sebuah obyek hanya akan dapat divisualisasikan pada komputer dalam bentuk diskrit. Oleh karena itu banyak metode yang khusus digunakan untuk mengubah bentuk kontinyu dari obyek menjadi bentuk diskrit. Metode yang biasa digunakan adalah *finite difference* dan *finite element*.

Finite difference merupakan metode yang membagi domain material menjadi kisi-kisi dan kemudian diterapkan diferensial numerik sebagai pendekatan spasial yang nantinya akan digunakan untuk mendapatkan nilai *strain* dan *strain tensor*. Pendekatan ini cocok untuk permasalahan yang memiliki struktur reguler atau teratur

Finite element merupakan metode yang membagi domain menjadi sub-domain atau elemen-elemen. Pada setiap elemen, material akan dipisahkan sesuai dengan fungsi yang dibatasi beberapa parameter. Fungsi-fungsi tersebut diuraikan ke dalam beberapa bagian seperti bentuk ortogonal, atau basis, fungsi yang berhubungan dengan batas *node* pada elemen yang ada. Elemen yang berdekatan akan memiliki *node* yang sama sehingga jejaring yang didefinisikan pada fungsi yang sama merepresentasikan domain material (O'Brien & Hodgins, 1999).

2.2.1. eXtended Finite Element Method

XFEM atau eXtended Finite Element Method dikonsepsikan dengan tujuan untuk membangun model sobekan dan aliran sobekan tanpa perlu *remeshing* atau penyesuaian *mesh* akibat adanya perubahan baik posisi maupun ukuran *mesh*. Pembentukan sobekan tanpa *remeshing* termasuk dari elemen-elemen dengan bentuk kontinu, teknik pergerakan *mesh*, dan teknik enrichment yang berbasis partition-of-unity yang mencakup remeshing dengan tingkat yang minimal.

Ide utama dari metode pada XFEM ini adalah menambahkan fungsi pengayaan atau *enrichment function* sebagai pendekatan yang mengandung

perpindahan elemen-elemen akibat terjadinya sebuah ketidaksinambungan atau diskontinuitas.

Pada konsep elastostatis, perubahan atau perpindahan kecil yang terjadi pada *mesh* direpresentasikan sebagai sebuah persamaan equilibrium yang dinotasikan dengan Persamaan 2.1.

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \text{ in } \Omega, \quad \boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} = \nabla_s \mathbf{u} \quad (2.1)$$

Pada Persamaan, $\Omega \subset \mathbf{R}^2$ adalah domain dari objek, $\boldsymbol{\sigma}$ adalah *stress tensor* Cauchy, $\boldsymbol{\varepsilon}$ adalah small strain tensor, \mathbf{b} adalah gaya internal objek, \mathbf{C} adalah tensor dari material, \mathbf{u} adalah perpindahan, ∇ adalah operator gradien, dan ∇_s adalah operator gradien simetris.

Sedangkan secara umum, pendekatan dari sebuah perpindahan atau *displacement* pada XFEM diformulasikan sebagai Persamaan 2.2.

$$u^h(x) = \sum_{I \in \Omega} N_I(x) \left[u_I + \sum_{I \in \Omega_d} v(x) a_I \right] \quad (2.2)$$

Variabel Ω pada Persamaan 2.2 merupakan keseluruhan domain sementara Ω_d merupakan domain yang mengalami diskontinuitas. Variabel $N_I(x)$ adalah fungsi Finite Element tradisional, $v(x)$ adalah fungsi diskontinuitas elemen pengayaan, sedangkan u_I dan a_I merupakan *degree of freedom* dari masing-masing Finite Element tradisional dan elemen pengayaan. Saat terjadi kondisi $\Omega \cap \Omega_d = \emptyset$, fungsi pengayaan $v(x)$ akan dihilangkan. Karena diskontinuitas tidak didefinisikan sebagai *mesh*, maka diskontinuitas ini didefinisikan ke dalam sebuah Level Set.

Pendekatan pada Persamaan 2.2 tidak mencakup sifat interpolasi akibat adanya *degree of freedom* dari elemen pengayaan. Untuk mengatasi hal ini, dibentuk formulasi fungsi pengayaan seperti Persamaan 2.3.

$$Y_I(x) = v(x) - v_I(x) \quad (2.3)$$

Variabel $Y_I(x)$ pada Persamaan 2.3 adalah fungsi elemen pengayaan yang ‘bergerak’ pada *node* ke- i dan $v_I(x)$ adalah nilai dari $v(x)$ pada *node* ke- i . Akibat adanya Persamaan ini nilai interpolasi akan terbentuk dan ketika penghitungan

formulasi mencapai elemen pengayaan, maka $Y_I(x)$ akan bernilai nol pada *node* tersebut.

Implementasi Persamaan 2.3 pada Persamaan 2.2 akan menghasilkan Persamaan 2.4 yang merepresentasikan pendekatan dari perpindahan yang terjadi pada *mesh*.

$$u^h(x) = \sum_I N_I(x) \left[u_I + \sum_J Y_I^J(x) a_I^J \right] \quad (2.4)$$

Pada Persamaan 2.4, $Y_I^J(x)$ adalah fungsi dari elemen pengayaan yang bergeser pada iterasi ke- J dan *node* ke- I .

2.2.2. Elemen Pengayaan

Metode Finite Element digunakan untuk menyelesaikan permasalahan yang berkaitan dengan diskontinuitas. Pada metode Finite Element, terutama XFEM, terdapat fungsi tambahan yang digunakan untuk memodelkan sobekan, kekosongan atau *voids*, dan inhomogenitas. Fungsi-fungsi tambahan ini diaplikasikan untuk meningkatkan akurasi pada bidang-bidang yang mementingkan penyesuaian sifat. (Sukumar, et al., 2001)

Pada sebuah model sobekan, dapat ditentukan diskontinuitas terjadi pada perpindahan yang berkaitan dengan *mesh*, tegangan yang terjadi, atau pada keduanya. Cara terbaik untuk dapat menunjukkan sebuah diskontinuitas adalah dengan membuat satu atau beberapa elemen sebagai penanda. Pada XFEM, diperkenalkan penggunaan *hanging nodes* yang berkaitan dengan sobekan, inklusi, dan aliran mekanik agar dapat dilakukan proses *mesh refinement* secara otomatis pada titik-titik diskontinu (Fries, et al., 2011).

a. Pengayaan Titik Inklusi

Titik inklusi merupakan salah satu representasi *node* yang bersifat homogenitas yang menyatakan batasan internal atau batasan yang berasal dari bidang objek itu sendiri. Pengayaan titik inklusi ini pertama kali disesuaikan dengan kebutuhan metode Level Set oleh Sukumar (2001). Pengayaan ini diformulasikan dengan Persamaan 2.5.

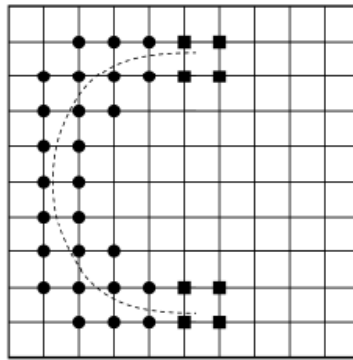
$$v(x) = \left| \sum_I N_I(x) \zeta_I \right| \quad (2.5)$$

Variabel ζ_I pada Persamaan 2.5 adalah nilai *node* pada nilai Level Set. Namun penyelesaian dengan Persamaan 2.5 dapat menimbulkan masalah *blending* pada elemen dengan pengayaan dan tanpa pengayaan. Untuk meningkatkan tingkat konvergensi dari penyelesaian permasalahan ini, maka dilakukan optimasi. Optimasi ini menerapkan modifikasi nilai absolut dari nilai pengayaan (Moes, et al., 2003). Dilakukan formulasi ulang sehingga Persamaan 2.5 dapat sesuai dengan formulasi elemen pengayaan menjadi Persamaan 2.6. Fungsi pengayaan ini akan bernilai nol pada *node* yang tidak perlu melakukan perpindahan.

$$Y(x) = \sum_I N_I(x) |\zeta_I| - |\sum_I N_I(x) \zeta_I| \quad (2.6)$$

b. Pengayaan Titik Retak dan Heaviside

Sebuah *displacement* atau perpindahan *node* tertentu bisa didapatkan dengan menggabungkan dua fungsi yang berkaitan dengan pendekatan sobekan itu sendiri. Dua fungsi tersebut adalah fungsi elemen *crack tip* atau titik ujung sobekan dan fungsi elemen Heaviside. Perbedaan kedua elemen ini berada pada fungsi yang dilakukan dan titik yang dipengaruhi.



Gambar 2.1 Node-node elemen pengayaan. Fungsi Heaviside ditunjukkan dengan titik bulat dan titik sobekan ditunjukkan dengan titik kotak (Pais, 2011)

Fungsi elemen Heaviside digunakan untuk merepresentasikan sobekan yang mengalir dari titik ujung sobekan. Sebuah titik ini dinyatakan termasuk elemen Heaviside apabila memenuhi Persamaan 2.7.

$$h(x) = \begin{cases} 1 & , \text{di atas titik sobek} \\ -1 & , \text{di bawah titik sobek} \end{cases} \quad (2.7)$$

Untuk elemen titik ujung sobekan, terdapat fungsi dengan empat nilai pengayaan yang digunakan untuk menyatakan perpindahan sobekan dari titik ujung sobekan ke wilayah selanjutnya. Fungsi tersebut dinyatakan dengan Persamaan 2.8.

$$\phi_{\alpha}(x)_{,\alpha=1-4} = \sqrt{r} \left[\sin \frac{\theta}{2}, \cos \frac{\theta}{2}, \sin \theta \sin \frac{\theta}{2}, \sin \theta \cos \frac{\theta}{2} \right] \quad (2.8)$$

Variabel r dan θ menyatakan koordinat polar pada sistem koordinat ujung sobekan lokal. Koordinat ini bermula dari titik ujung sobekan dan $\theta = 0$ paralel dengan sobekan.

Persamaan 2.8 akan menunjukkan diskontinuitas yang terjadi saat ujung sobekan terbentuk sementara Persamaan 2.7 akan menunjukkan potongan yang terjadi akibat munculnya sobekan. Pengaruh kedua fungsi ini ditunjukkan pada Gambar 2.1.

Perhitungan kedua fungsi ini belum menyatakan sobekan pada domain objek yang dituju. Oleh karena itu dibutuhkan metode yang merepresentasikan fungsi-fungsi ini ke domain yang diinginkan sehingga diterapkan metode Level Set. Terdapat dua Level Set yang dibutuhkan untuk merepresentasikan sobekan. Pertama adalah level set dasar $\psi(x)$ yang merepresentasikan fisik sobekan. Yang kedua adalah level set dasar $\phi(x)$ yang ortogonal dengan level set dasar $\psi(x)$. Persamaan 2.7 dan 2.8 dapat dikalkulasi mengikuti bentuk $\phi(x)$ dan $\psi(x)$ sehingga didapatkan Persamaan 2.9.

$$h(x) = h(\psi(x)) = \begin{cases} 1 & \text{untuk } \psi(x) > 0 \\ -1 & \text{untuk } \psi(x) < 0 \end{cases} \quad (2.9)$$

Sedangkan koordinat polar didapatkan dengan Persamaan 2.10 dan 2.11.

$$r = \sqrt{\psi^2(x) + \phi^2(x)} \quad (2.10)$$

$$\theta = \arctan \frac{\psi(x)}{\phi(x)} \quad (2.11)$$

Node-node pengayaan yang berhubungan dengan titik ujung sobekan juga dapat ditentukan dengan menggunakan Level Set yang mendefinisikan sobekan. *Node* Heaviside dapat ditentukan dengan Persamaan 2.12. Sedangkan *Node* titik ujung sobekan dapat ditentukan ketika memenuhi Persamaan 2.13.

$$\phi_{max} < 0 \text{ dan } \psi_{max}\psi_{min} \leq 0 \quad (2.12)$$

$$\phi_{max}\phi_{min} \leq 0 \text{ dan } \psi_{max}\psi_{min} \leq 0 \quad (2.13)$$

Setelah diterapkan titik ujung sobekan dan Heaviside, perubahan yang terjadi pada domain seperti yang dinyatakan pada Persamaan 2.4 akan mengalami penyesuaian. Penyesuaian ini dilakukan dengan menerapkan metode Bubnov-Galerkin untuk membangun sistem persamaan dan menurunkan Persamaan 2.4 sehingga dapat merepresentasikan domain secara total. Pendekatan ini dinyatakan dalam Persamaan 2.14.

$$u^h(x) = \sum_I N_I(x) \left[u_I + H_I(x)a_I + \sum_{a=1}^4 \Phi_a(x)b_I^a \right] \quad (2.14)$$

Variabel $H_I(x)$ adalah elemen Heaviside dan $\Phi_a(x)$ adalah titik ujung sobekan.

2.2.3. Metode Level Set

Metode Level Set adalah metode numerik yang digunakan untuk menelusuri evolusi atau perubahan posisi yang terjadi pada antarmuka atau obyek yang tampak oleh pengguna. Metode ini diterapkan berdasarkan kecepatan fisik yang terjadi pada obyek sehingga menimbulkan perpindahan posisi *node* pada setiap iterasi berdasarkan waktu. Kecepatan yang ada pada antarmuka objek dapat dipengaruhi oleh posisi, waktu, geometri bidang, dan fisik yang ada pada model.

Metode Level Set digunakan untuk menentukan nilai diskrit dari domain awal objek menjadi titik-titik diskrit. Selain itu, Level Set digunakan untuk merepresentasikan antarmuka bidang sebagai serat atau *fiber* dan matrik sehingga dapat merepresentasikan bentuk geometri dari sobekan.

Fungsi Level Set dinyatakan dengan $\phi(x(t), t)$. Sesuai dengan formulasi tersebut, fungsi ini merupakan fungsi kontinu dengan variabel $x(t)$ adalah titik pada sebuah domain awal Ω . Batasan yang diterapkan untuk memenuhi fungsi Level Set diformulasikan pada Persamaan 2.15, 2.16, dan 2.17. Variabel Γ pada Persamaan 2.17 adalah representasi domain antarmuka.

$$\phi(x(t), t) < 0 \text{ untuk } x \in \Omega \quad (2.15)$$

$$\phi(x(t), t) > 0 \text{ untuk } x \in \Omega \quad (2.16)$$

$$\phi(x(t), t) = 0 \text{ untuk } x \in \Gamma \quad (2.17)$$

Domain antarmuka dapat ditemukan pada sembarang waktu t dengan lokasi yang sesuai dengan syarat $\phi(x(t), t) = 0$ seperti pada Persamaan 2.17. Batasan ini biasa disebut *zero level set* ϕ dan bisa disimbolkan sebagai ϕ_0 . Persamaan 2.17 inilah yang dinamakan persamaan Level Set.

Untuk mendapatkan evolusi perubahan posisi objek untuk setiap satuan waktu, dilakukan pendekatan dengan melakukan penurunan rumus persamaan Level Set terhadap waktu t . Penurunan ini dinyatakan dengan Persamaan 2.18.

$$\frac{\phi_{i+1} - \phi_i}{\Delta t} + V_i \cdot \nabla \phi_i = 0 \quad (2.18)$$

Variabel ϕ_{i+1} pada Persamaan 2.18 adalah nilai Level Set pembaharuan, variabel ϕ_i adalah nilai Level Set pada titik yang sedang diproses, V_i adalah vektor kecepatan, dan Δt adalah perbedaan waktu yang terjadi antara i dan $i+1$. Persamaan 2.18 dapat ditulis kembali ke dalam bentuk yang lebih sesuai dengan objek dua dimensi pada Persamaan 2.19.

$$\phi_{i+1} = \phi_i - \Delta t \left(u_i \frac{\partial \phi_i}{\partial x} + v_i \frac{\partial \phi_i}{\partial y} \right) \quad (2.19)$$

Variabel u_i adalah kecepatan terhadap arah x dan v_i adalah kecepatan terhadap arah y . Pada Persamaan 2.18 dan 2.19, variabel Δt dibatasi oleh kondisi Courant-Friedrichs-Lewy (CFL) yang memastikan bahwa pendekatan dari solusi konvergen. Kondisi CFL dinyatakan dengan Persamaan 2.20. Variabel Δx dan Δy pada Persamaan 2.20 adalah nilai x dan y pada *grid* atau representasi matrik dasar dari objek.

$$\Delta t < \frac{\max(\Delta x, \Delta y)}{\max(u, v)} \quad (2.20)$$

Metode Level Set juga diterapkan pada elemen pengayaan. Untuk nilai elemen pengayaan dengan bentuk material komposit yang berbentuk lingkaran, fungsi Level Set yang diterapkan pada setiap node dinyatakan dalam fungsi $\zeta(x)$ seperti pada Persamaan 2.21. Pada persamaan ini, x_i dan y_i merupakan koordinat titik ke- i dan x_c dan y_c merupakan koordinat titik pusat material komposit, sedangkan r_c adalah jari-jari lingkaran tersebut.

$$\zeta(x) = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r_c \quad (2.21)$$

Sedangkan untuk elemen pengayaan yang berkenaan dengan sobekan—titik ujung sobekan dan Heaviside, dihitung jarak antara setiap titik dengan *zero level set* yang terdekat. Retakan terdefinisi apabila persyaratan pada Persamaan 2.22 dan 2.23 terpenuhi.

$$\phi(x(t), t) \leq 0 \quad (2.22)$$

$$\psi(x(t), t) = 0 \quad (2.23)$$

2.3. Stiffness Matrix

Nilai perpindahan yang dihitung dengan Persamaan 2.14 terjadi pada wilayah komputasi numerik. Untuk mengaplikasikan perpindahan agar dapat direpresentasikan pada domain baik domain *mesh* maupun domain antarmuka, dilakukan *convert* nilai perpindahan agar sesuai dengan bentuk nodalnya. Hal ini dapat dilakukan dengan menggunakan metode Bubnov-Galerkin yang dapat membentuk perpindahan ke dalam bentuk persamaan linier. Persamaan linier ini dinyatakan dalam Persamaan 2.24.

$$\mathbf{K} \cdot \mathbf{q} = \mathbf{f} \quad (2.24)$$

Variabel \mathbf{K} merupakan stiffness matrix global, \mathbf{q} adalah *degree of freedom* dari *node*, dan \mathbf{f} adalah gaya yang berlaku pada *node*. Sesuai dengan urutan *degree of freedom*, global stiffness matrix dapat dinyatakan dengan Persamaan 2.25.

$$\mathbf{K} = \begin{bmatrix} K_{uu} & K_{ua} & K_{ub} \\ K_{ua}^T & K_{aa} & K_{ab} \\ K_{ub}^T & K_{ab}^T & K_{bb} \end{bmatrix} \quad (2.25)$$

Variabel u pada Persamaan 2.25 menyatakan nilai kekakuan pada Finite Element tradisional, sementara a menyatakan nilai pengayaan pada elemen Heaviside, dan b menyatakan nilai pengayaan pada titik ujung sobekan.

Sesuai dengan perumusan Persamaan 2.25 terhadap Persamaan 2.24, dapat dilihat bahwa kekakuan pada sobekan dikerjakan oleh elemen Heaviside dan titik ujung sobekan sehingga elemen kekakuan pada Finite Element tradisional K_{uu} akan konstan selama iterasi penambahan titik retak. Matrik kekakuan juga menyatakan bahwa porsi perubahan yang diakibatkan elemen pengayaan akan lebih kecil daripada elemen non-pengayaan. Lebih jauh, dapat dilihat bahwa setelah sebuah elemen dikenakan Heaviside, nilai kekakuannya tidak akan berubah pada iterasi-

iterasi berikutnya. Hal ini berarti komponen kekakuan yang mengandung a akan konstan selama iterasi pembentukan sobekan terjadi.

Oleh karena itu, matrik kekakuan dapat disederhanakan untuk dapat menghitung evolusi pertumbuhan sobekan dan menghasilkan elemen baru berupa titik ujung sobekan pada iterasi-iterasi yang terjadi setelah kondisi inisial. Matrik kekakuan global yang digunakan dinyatakan dalam Persamaan 2.26.

$$K = \begin{bmatrix} K_{uu} & K_{ua} \\ K_{ua}^T & K_{aa} \end{bmatrix} \quad (2.26)$$

Berdasarkan matrik dasar dengan Persamaan 2.26, bisa didapatkan nilai kekakuan elemental untuk setiap anggota K . Kekakuan elemental ini dihitung dengan Persamaan 2.27.

$$K_e = \int_{\Omega^h} B_u^T C B_a d\Omega \quad (2.27)$$

Variabel C adalah matrik konstitutif untuk material dengan *isotropic linear elastic*. Variabel B_u adalah matrik turunan dari fungsi matrik domain awal sedangkan B_a adalah matrik turunan dari matrik elemen pengayaan. Matrik B_u dan B_a dijabarkan pada Persamaan 2.28 dan 2.29.

Variabel $N_{I,i}$ pada Persamaan 2.28 adalah turunan dari $N_I(x)$ terhadap x_i . Sedangkan variabel $(N_I Y_I^J)_{,i}$ pada Persamaan 2.29 merupakan turunan $N_I(x) Y_I^J$ terhadap x_i dengan Y adalah elemen pengayaan.

$$B_u = \begin{bmatrix} N_{I,x} & 0 & 0 \\ 0 & N_{I,y} & 0 \\ 0 & 0 & N_{I,z} \\ 0 & N_{I,z} & N_{I,y} \\ N_{I,z} & 0 & N_{I,x} \\ N_{I,y} & N_{I,x} & 0 \end{bmatrix} \quad (2.28)$$

$$B_a = \begin{bmatrix} (N_I Y_I^J)_{,x} & 0 & 0 \\ 0 & (N_I Y_I^J)_{,y} & 0 \\ 0 & 0 & (N_I Y_I^J)_{,z} \\ 0 & (N_I Y_I^J)_{,z} & (N_I Y_I^J)_{,y} \\ (N_I Y_I^J)_{,z} & 0 & N_{I,x} \\ (N_I Y_I^J)_{,y} & (N_I Y_I^J)_{,x} & 0 \end{bmatrix} \quad (2.29)$$

Sesuai dengan Persamaan 2.24, formulasi akan lengkap dengan perhitungan *degree of freedom*. Perhitungan *degree of freedom* dilakukan dengan Persamaan 2.30. Pada Persamaan 2.30, variabel u dan a adalah *degree of freedom* untuk elemen tradisional dan elemen pengayaan.

$$q^T = \{u \quad a\}^T \quad (2.30)$$

Setelah matrik kekakuan dan *degree of freedom* didapatkan, maka tegangan dan tekanan atau *strain* dan *stress* dapat dihitung dengan Persamaan 2.31 dan 2.32.

$$\varepsilon = [B_u \quad B_a] \{u \quad a\}^T \quad (2.31)$$

$$\sigma = C \varepsilon \quad (2.32)$$

2.4. Model Sobekan dengan J-Integral

Metode Finite Element mengatur dan memperhitungkan segala komputasi numerik mulai dari implementasi objek ke dalam bentuk matrik, proses diskrit objek, hingga penggabungan elemen-elemen yang mempengaruhi objek untuk dapat membentuk sobekan. Pada bab ini akan dijelaskan mengenai model s yang akan mencakup proses perhitungan nilai J sebagai energi yang dilepaskan yang menentukan arah dan besar sobekan yang terjadi.

Sudut sobekan dihitung dengan Persamaan 2.33 dengan K_I dan K_{II} adalah faktor intensitas tegangan atau *stress intensity factor*

$$\theta_c = 2 \arctan \frac{1}{4} \left(\frac{K_I}{K_{II}} - \text{sign}(K_{II}) \sqrt{\left(\frac{K_I}{K_{II}} \right)^2 + 8} \right) \quad (2.33)$$

Arah dari pertumbuhan sobekan ditentukan pula oleh mode-gabungan (*mixed-mode*) faktor intensitas tegangan. Bentuk domain dari interaksi integral digunakan untuk menghitung mode-gabungan faktor intensitas tegangan. Untuk

metode-gabungan secara umum, hubungan antara J-Integral dan faktor intensitas tegangan dinyatakan dengan Persamaan 2.34.

$$J = \frac{K_I^2}{E_{eff}} + \frac{K_{II}^2}{E_{eff}} \quad (2.34)$$

Variabel E_{eff} didefinisikan dengan pertidaksamaan dari plane stress dan plane strain seperti Pertidaksamaan 2.35 dengan E adalah modulus Young dan ν adalah rasio Poisson.

$$E_{eff} = \begin{cases} E, & \text{plane stress} \\ \frac{E}{1-\nu^2}, & \text{plane strain} \end{cases} \quad 2.35 \quad (2.35)$$

Untuk menghitung mode-gabungan faktor intensitas tegangan, diberikan sebuah kondisi tegangan sebagai bantuan. Bantuan ini ditumpangkan pada tegangan dan perpindahan dari analisis XFEM. Elemen dari XFEM dinotasikan dengan tanda (1) seperti $\sigma_{ij}^{(1)}$, $\varepsilon_{ij}^{(1)}$, dan $u_i^{(1)}$, sementara elemen dari kondisi bantuan dinyatakan dengan $\sigma_{ij}^{(2)}$, $\varepsilon_{ij}^{(2)}$, dan $u_i^{(2)}$.

Kondisi awal J-Integral dinyatakan dengan Persamaan 2.36.

$$J_i = \int_{\Gamma} \left[W_{n_i} - \sigma_{jk} n_j \frac{\partial u_k}{\partial x_i} \right] d\Gamma \quad (2.36)$$

Variabel W pada Persamaan 2.36 adalah *strain energy density* dan i menotasikan arah bukaan sobekan, yang diasumsikan berkorespondensi dengan arah x secara global, yang dinotasikan dengan x_i . Persamaan 2.36 dapat dituliskan kembali sebagai Persamaan 2.37.

$$J_i = \int_{\Gamma} \left[W \delta_{1j} - \sigma_{ij} \frac{\partial u_i}{\partial x_1} \right] n_j d\Gamma \quad (2.37)$$

Kedua kondisi tegangan σ dapat ditumpangkan pada Persamaan 2.37 sehingga membentuk Persamaan 2.38.

$$J_1^{(1+2)} = \int_{\Gamma} \left[\frac{1}{2} (\sigma_{ij}^{(1)} + \sigma_{ij}^{(2)}) (\varepsilon_{ij}^{(1)} + \varepsilon_{ij}^{(2)}) \delta_{1j} - (\sigma_{ij}^{(1)} + \sigma_{ij}^{(2)}) \frac{\partial (u_i^{(1)} + u_i^{(2)})}{\partial x_1} \right] n_j d\Gamma \quad (2.38)$$

Bentuk J-Integral dengan kondisi 1 dan 2 yang terpisah dari Persamaan 2.38 dapat dirumuskan dengan Persamaan 2.39.

$$J_1^{(1+2)} = J_1^{(1)} + J_1^{(2)} + I^{(1,2)} \quad (2.39)$$

Variabel $I^{(1,2)}$ pada Persamaan 2.39 merupakan kondisi interaksi dan dihitung dengan Persamaan 2.40.

$$I^{(1,2)} = \int_{\Gamma} \left[W^{(1,2)} \delta_{1j} - \sigma_{ij}^{(1)} \frac{\partial u_i^{(2)}}{\partial x_1} - \sigma_{ij}^{(2)} \frac{\partial u_i^{(1)}}{\partial x_1} \right] n_j d\Gamma \quad (2.40)$$

Variabel $W^{(1,2)}$ pada Persamaan 2.40 adalah interaksi dari *strain energy density* yang dinyatakan dengan Persamaan 2.41.

$$W^{(1,2)} = \sigma_{ij}^{(1)} \varepsilon_{ij}^{(2)} = \sigma_{ij}^{(2)} \varepsilon_{ij}^{(1)} \quad (2.41)$$

Setelah menumpangkan kedua kondisi, Persamaan 2.34 dapat diformulasi ulang dengan Persamaan 2.42.

$$J_1^{(1+2)} = \frac{(K_I^{(1)} K_I^{(2)})^2}{E_{eff}} + \frac{(K_{II}^{(1)} K_{II}^{(2)})^2}{E_{eff}} \quad (2.42)$$

Pengembangan dan pengaturan ulang dilakukan untuk Persamaan 2.42 sehingga menjadi Persamaan 2.43.

$$J_1^{(1+2)} = J_1^{(1)} + J_1^{(2)} + \frac{2(K_I^{(1)} K_I^{(2)} + K_{II}^{(1)} K_{II}^{(2)})}{E_{eff}} \quad (2.43)$$

Faktor intensitas tegangan untuk kondisi yang sedang difokuskan dapat dihitung dengan memisahkan kedua mode sobekan. Dengan memilih $K_I^{(2)} = 1$ dan $K_{II}^{(2)} = 0$, didapatkan penyelesaian $K_I^{(1)}$ dengan Persamaan 2.44 dan penyelesaian $K_{II}^{(1)}$ dengan Persamaan 2.45.

$$K_I^{(1)} = \frac{I^{(1, Mode I)} E_{eff}}{2} \quad (2.44)$$

$$K_{II}^{(1)} = \frac{I^{(1, Mode II)} E_{eff}}{2} \quad (2.45)$$

[halaman ini sengaja dikosongkan]

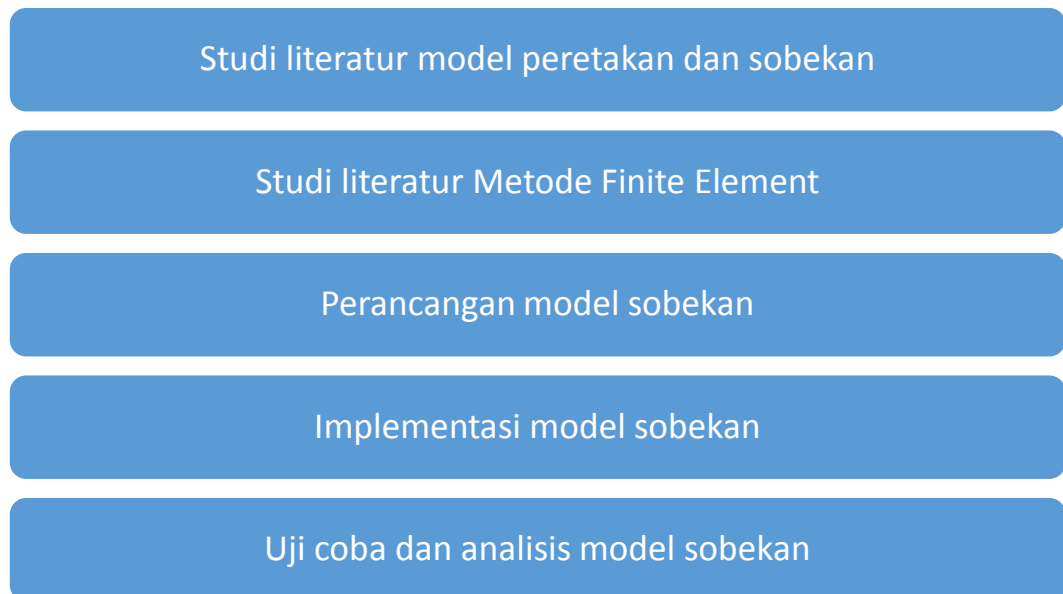
BAB 3

METODOLOGI PENELITIAN

Bab ini memaparkan metodologi penelitian yang akan digunakan. Pada bagian ini akan dijelaskan alur dari metodologi penelitian yang terdiri atas studi literatur, perancangan metode, implementasi metode, pengujian, dan analisa hasil uji coba. Selanjutnya dijelaskan rancangan sistem dan detailnya.

3.1. Alur Metode Penelitian

Penelitian ini terdiri atas beberapa tahap. Alur metodologi yang dilakukan pada penelitian ini dijabarkan dalam Gambar 3.1.



Gambar 3.1 Alur metodologi penelitian

3.1.1. Studi Literatur Model Peretakan dan Sobekan

Studi literatur tentang model peretakan dan sobekan dilakukan untuk dapat memahami model retakan dan sobekan yang selama ini telah dikerjakan oleh peneliti-peneliti lain. Pada studi literatur ini dilakukan pemahaman tentang dasar model yang harus dikerjakan. Dasar model ini meliputi pendalaman metode untuk merepresentasikan objek ke dalam bentuk mesh, pendalaman metode yang digunakan untuk mengekstraksi komponen-komponen objek sehingga dapat

dikomputasi, dan pendalaman formulasi-formulasi yang harus digunakan untuk membentuk model yang sesuai dengan tujuan penelitian.

3.1.2. Studi Literatur Metode Finite Element

Studi literatur tentang metode Finite Element dilakukan setelah pendalaman tentang objek dan representasi mesh dilakukan. Metode ini merupakan metode dasar yang digunakan untuk merepresentasikan bentuk nyata objek yang merupakan bentuk kontinu menjadi bentuk diskrit sehingga komponen-komponennya dapat dikomputasi oleh sistem. Metode ini memerlukan pemahaman yang lebih karena pada setiap kasus pembentukan model grafis, Finite Element yang diterapkan menyesuaikan dengan bentuk geometri serta permasalahan yang harus diselesaikan. Literatur-literatur yang mendukung dikumpulkan dan diproses sehingga membentuk model yang sesuai.

3.1.3. Perancangan Model Sobekan

Perancangan model sobekan dilakukan untuk mengolaborasikan metode-metode yang cocok menjadi sebuah sistem yang dapat merepresentasikan objek dengan semirip mungkin. Metode-metode mulai dari pengimplementasian objek ke dalam bentuk *mesh*, pengubahan bentuk kontinu ke dalam bentuk diskrit, implementasi bentuk plastisitas, dan implementasi retakan yang dihasilkan harus dapat dikolaborasikan agar dapat berjalan dengan baik.

3.1.4. Implementasi Model Sobekan

Implementasi dilakukan dengan membangun model secara keseluruhan. Model dibentuk sesuai dengan kebutuhan dengan masukan-masukan yang disesuaikan dengan kasus penelitian.

Tahapan ini dilakukan untuk mengimplementasikan metode yang diusulkan ke dalam perangkat lunak. Proses implementasi ini meliputi penulisan kode program, *compiling*, *debugging*, hingga dihasilkan kode program yang siap dieksekusi. Tahapan ini dilakukan setelah dilakukan pembelajaran terhadap *library* yang digunakan untuk membentuk komponen-komponen yang dibutuhkan pada sebuah komputasi grafis.

3.1.5. Uji Coba dan Analisis Model Sobekan

Uji coba yang dilakukan terdiri atas uji coba untuk melihat keberhasilan model membentuk sobekan pada objek penelitian. Selain itu dilakukan untuk mengetahui variabel-variabel yang berpengaruh terhadap terbentuknya sobekan.

3.2. Rancangan Sistem

Tahap perancangan sistem ini merupakan implementasi dari metode-metode yang telah dipelajari untuk membentuk model sobekan. Tahapan-tahapan yang harus dilakukan untuk membentuk model sobekan dijabarkan dalam diagram alir yang ditunjukkan pada Gambar 3.2. Sesuai dengan Gambar 3.2, terdapat beberapa tahapan yang harus dieksekusi pada setiap iterasi. Tahapan-tahapan ini akan dijelaskan secara rinci pada subbab Rancangan Sistem.

3.2.1. Masukan

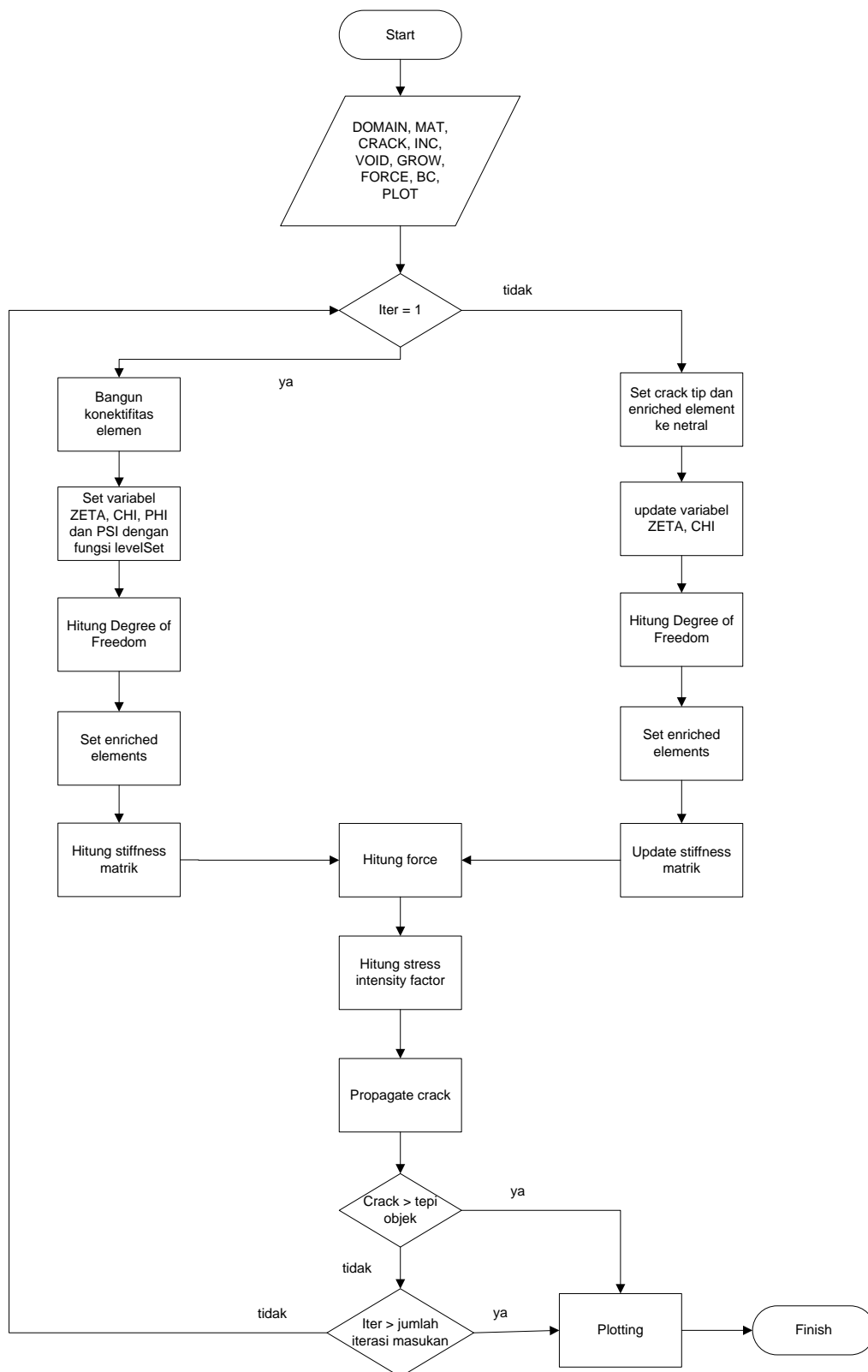
Masukan pengguna dikategorikan ke dalam beberapa komponen. Komponen ini adalah DOMAIN, MAT, CRACK, INC, GROW, dan FORCE. Komponen masukan DOMAIN merupakan sebuah variabel dengan empat nilai yang menyatakan:

1. Panjang objek pada arah x
2. Lebar objek pada arah y
3. Ukuran setiap elemen x
4. Ukuran setiap elemen y

Variabel kedua adalah MAT yang melingkupi tujuh nilai. Nilai-nilai ini menyatakan:

1. modulus Young domain,
2. rasio Poisson domain,
3. modulus Young titik inklusi,
4. rasio Poisson titik inklusi,
5. status stress-strain,
6. nilai plane stress,
7. faktor intensitas tegangan kritis pada domain.

Variabel ketiga adalah variabel CRACK. Variabel ini menyatakan garis retakan pertama yang terjadi pada domain. Pada variabel ini dinyatakan:



Gambar 3.2 Diagram Alir Model Sobekan

1. koordinat x titik pertama
2. koordinat y titik pertama
3. koordinat x titik kedua,
4. koordinat y titik kedua.

Variabel keempat adalah variabel INC yang menyatakan inklusi. Bila masukan pada INC berisi tiga nilai, maka nilai tersebut akan menyatakan inklusi lingkaran dengan keterangan:

1. koordinat x titik pusat inklusi,
2. koordinat y titik pusat inklusi,
3. jari-jari inklusi.

Sementara bila variabel INC berisi empat nilai, maka nilai tersebut menyatakan inklusi linier yang dinyatakan dengan:

1. koordinat x titik pertama,
2. koordinat y titik pertama ,
3. koordinat x titik kedua,
4. koordinat y titik kedua.

Variabel berikutnya adalah variabel GROW yang nantinya akan mempengaruhi besar pertumbuhan sobekan pada domain. Variabel ini menyatakan:

1. jumlah iterasi,
2. *crack growth increment*.

Variabel terakhir adalah variabel FORCE yang akan menentukan muatan yang dialirkan pada domain. Variabel ini menyatakan tiga keterangan:

uniaxial tension

nilai gaya yang diaplikasikan pada arah x

nilai gaya yang diaplikasikan pada arah y

3.2.2. Fungsi Utama

Fungsi utama diterapkan sebagai fungsi eksekusi dari keseluruhan program. Fungsi ini mengandung skenario eksekusi fungsi-fungsi lain yang ada pada program. Fungsi ini merupakan representasi kode dari Gambar 3.2 dan dinyatakan dengan *pseudocode* pada Gambar 3.3. Fungsi-fungsi lain yang dieksekusi pada fungsi ini dijelaskan pada subbab-subbab berikutnya

<i>Pseudocode 3.1 Fungsi utama</i>	
Masukan	-
Keluaran	PLOT, waktu
<pre> 1. input 2. if isempty(GROW) = 1 3. iter ← 1 4. elseif length(GROW) = 2 5. iter ← GROW(1) 6. end 7. for i = 1 to iter 8. if i = 1 9. connectivity 10. pHDOF ← [] 11. else 12. clear status on NODES 13. end 14. omega ← levelSet(i) 15. [DOF,DISP] ← calcDOF 16. [updElem, IElem] = enrElem(i,pHDOF) 17. 18. if i = 1 19. globalK ← stiffnessMatrix 20. else 21. globalK ← updateStiffness 22. end 23. globalF ← forceVector 24. freeDOF ← boundaryCond 25. DISP(freeDOF,:) ← globalK \ globalF 26. if isempty(CRACK) != 0 27. pHDOF ← 2*max(NODES(:,2)) 28. [KI,KII] ← JIntegral(omega,DISP) 29. Gopt ← -(KI(1)^2+KII(1)^2)*(1-MAT(2)^2)/MAT(1); 30. growCrack(KI,KII,omega) 31. if i ~=1 32. if CRACK(nPt,1) >= (nXElem*lXElem-2*lXElem) % 33. CRACK(nPt,1) < CRACK(nPt-1,1) 34. plotMain(omega,DISP); break 35. end 36. end 37. if i > iter 38. plot result 39. break 40. end 41. print time 42. end </pre>	

Gambar 3.3. Pseudocode fungsi utama

3.2.3. Fungsi connectivity

Fungsi connectivity dapat dikatakan fungsi inisialisasi. Fungsi ini akan membentuk *grid* dari masukan yang telah diberikan pengguna. *Grid* inilah yang menjadi representasi objek. Fungsi connectivity akan membentuk *grid* dengan informasi masukan dan membentuk informasi-informasi lainnya seperti koordinat titik, keterhubungan antar titik, hingga elemen serta titik-titik yang menjadi anggotanya. Fungsi ini hanya dieksekusi pada iterasi pertama. Untuk iterasi berikutnya, informasi yang digunakan juga merupakan hasil dari fungsi ini namun dengan nilai-nilai yang telah diproses pada fungsi-fungsi lain.

Proses eksekusi fungsi connectivity dijelaskan dalam bentuk *pseudocode* pada Gambar 3.4.

<i>Pseudocode</i> 3.2 Konektivitas domain	
Masukan	DOMAIN
Keluaran	NODES, XYZ, CONNEC
<pre> 1. nXElem ← DOMAIN(1) 2. nYElem ← DOMAIN(2) 3. lXElem ← DOMAIN(3) 4. lYElem ← DOMAIN(4) 5. nNode = (nXElem+1)*(nYElem+1) 6. nNode ← 1; 7. for iYNode ← 1 to nYElem+1 8. for iXNode ← 1 to nXElem+1 9. NN(iXNode,iYNode) ← nNode 10. XYZ (nNode,:) ← [nNode (iXNode-1)*lXElem (iYNode-1)*lYElem] 11. NODES(nNode,1) ← nNode 12. nNode ← nNode+1 13. end 14. end 15. nElem ← 1; 16. for iYElem ← 1 to nYElem 17. for iXelem ← 1 to nXElem 18. N1 ← NN(iXElem, iYElem) 19. N2 ← NN(iXElem+1, iYElem) 20. N3 ← NN(iXElem+1,iYElem+1) 21. N4 ← NN(iXElem, iYElem+1) 22. CONNEC(nElem,1:5) ← [nElem N1 N2 N3 N4] 23. nElem ← nElem+1 24. end 25. end </pre>	

Gambar 3.4 Pseudocode Fungsi connectivity

3.2.4. Fungsi LevelSet

Fungsi levelSet merupakan fungsi yang membentuk nilai level set \emptyset yang merepresentasikan domain. Elemen dengan pengayaan titik ujung sobekan dan Heaviside ditentukan. Pada fungsi ini akan diidentifikasi titik retakan berdasarkan masukan pengguna. Selain itu, fungsi ini juga akan mengidentifikasi retakan yang terjadi pada iterasi lanjut dengan titik retakan hasil iterasi sebelumnya.

Fungsi ini juga akan membentuk level set PHI dan PSI yang menyatakan diskontinuitas yang disebabkan oleh titik ujung sobekan dan titik-titik Heaviside. Titik-titik ini kemudian akan memiliki nilai *degree of freedom* sesuai dengan pengayaan yang dimilikinya. Fungsi levelSet dieksekusi sesuai dengan *pseudocode* seperti pada Gambar 3.5, Gambar 3.6, Gambar 3.7, dan Gambar 3.8.

<i>Pseudocode 3.3. Inisialisasi elemen dengan metode Level Set</i>	
Masukan	iter, CONNEC, DOMAIN, INC, NODES, XYZ, CRACK
Keluaran	Omega, PHI, PSI, ZETA
<pre> 1. nXElem ← DOMAIN(1) 2. nYElem ← DOMAIN(2) 3. lXElem ← DOMAIN(3) 4. nElem ← nXElem*nYElem 5. nNode ← (nXElem+1)*(nYElem+1); 6. nPt ← size(CRACK,1); 7. nBand ← (sqrt(2)+0.005)*lXElem 8. //Create material level set function ZETA 9. if iter = 1 10. if INC != 0 11. if INC circular 12. xc ← INC(1) 13. yc ← INC(2) 14. rc ← INC(3) 15. for iNode = 1 to nNode 16. xi ← XYZ(iNode,2) 17. yi ← XYZ(iNode,3) 18. Z ← sqrt((xi-xc)^2+(yi-yc)^2)-rc 19. if abs(z) < 1e-6, ZETA(iNode) ← 0 20. else ZETA(iNode) ← Z 21. end 22. end 23. else if INC linear 24. x0 ← INC(1) 25. y0 ← INC(2) 26. x1 ← INC(3) 27. y1 ← INC(4) </pre>	

Gambar 3.5 Pseudocode Inisialisasi elemen dengan metode Level Set

<i>Pseudocode 3.4. Inisialisasi elemen dengan metode Level Set</i>	
Masukan	iter, CONNEC, DOMAIN, INC, NODES, XYZ, CRACK
Keluaran	Omega, PHI, PSI, ZETA
28.	$l \leftarrow \sqrt{(x1-x0)^2 + (y1-y0)^2}$
29.	for iNode = 1 to nNode
30.	xi \leftarrow XYZ(iNode,2)
31.	yi \leftarrow XYZ(iNode,3)
32.	c \leftarrow (y0-y1)*x+(x1-x0)*y+(x0*y1-x1*y0)
33.	ZETA(iNode) \leftarrow c/l
34.	end
35.	end ZETA \leftarrow min(ZETA, [], 1)
36.	for iElem = 1 to nElem
37.	Z1 \leftarrow ZETA(CONNEC(iElem,2))
38.	Z2 \leftarrow ZETA(CONNEC(iElem,3))
39.	Z3 \leftarrow ZETA(CONNEC(iElem,4))
40.	Z4 \leftarrow ZETA(CONNEC(iElem,5))
41.	Z \leftarrow [Z1 Z2 Z3 Z4]
42.	if max(Z)*min(Z) < 0for iNode = 2 to 5
43.	if NODES(CONNEC(iElem,iNode),30) == 0
44.	NODES(CONNEC(iElem,iNode),30) =
	nNode+incNode;
45.	NODES(CONNEC(iElem,iNode),31) = Z(iNode-
	1);
46.	incNode = incNode+1;
47.	end
48.	end
49.	end
50.	else
51.	ZETA \leftarrow ones(1,nNode)
52.	end
53.	//Create level set function PHI and PSI defining the crack
54.	if CRACK != empty
55.	nCT \leftarrow Check for edge crack
56.	Calculate crack angle
57.	PHI \leftarrow sparse(nNode,nCT)
58.	for iCT = 1: nCT
59.	define radius of nodal search
60.	find nodes within search radius
61.	//compute PHI level set for main cracktip
62.	for iNode = 1 to length of domain
63.	cNode \leftarrow domain(iNode)
64.	x \leftarrow XYZ(cNode,2)
65.	y \leftarrow XYZ(cNode,3)
66.	disc \leftarrow CRACK(nPt,:) - CRACK(nPt-1,:)
67.	t \leftarrow 1/norm(disc)*disc
68.	phi \leftarrow ([x y] - CRACK(nPt,:)) * t';

Gambar 3.6 Pseudocode Inisialisasi elemen dengan metode Level Set (2)

<i>Pseudocode 3.5</i> Inisialisasi elemen dengan metode Level Set	
Masukan	iter, CONNEC, DOMAIN, INC, NODES, XYZ, CRACK
Keluaran	Omega, PHI, PSI, ZETA
69.	if phi = 0, phi = 1e-6; end
70.	if nCT == 2
71.	disc \leftarrow CRACK(1,:) - CRACK(2,:)
72.	t \leftarrow 1/norm(disc)*disc
73.	phi(2) \leftarrow ([x y] - CRACK(1,:))*t';
74.	if phi(2) = 0, phi(2) = 1e-6; end
75.	end
76.	if iCT = 1
77.	if abs(phi(1)) < nBand
78.	define psi
79.	if abs(psi) < nBand
80.	PHI(cNode,1) = phi(1)
81.	PSI(cNode,1) = psi end
82.	else if phi(1) < 0
83.	define psi
84.	if abs(psi) < nBand
85.	PSI(cNode,1) = psi
86.	end
87.	end
88.	if iCT = 2
89.	if abs(phi(2)) < nBand
90.	define psi
91.	if abs(psi) < nBand
92.	PHI(cNode,1) = phi(2)
93.	PSI(cNode,1) = psi
94.	end
95.	else if phi(2) < 0
96.	define psi
97.	if abs(psi) < nBand
98.	PSI(cNode,1) = psi
99.	end
100.	end
101.	end
102.	end
103.	define the crack tip enriched nodes
104.	define Heaviside enriched nodes
105.	//Preliminaries to numbering enriched nodes
106.	nNode \leftarrow nNode+1+incNode-1
107.	ctNodes \leftarrow unique(ctNodes)
108.	bmNodes \leftarrow unique(bmNodes)
109.	hNodes \leftarrow unique(hNodes)
110.	// Number the Heaviside nodes
111.	for i = 1 to length of hNodes
112.	NODES(hNodes(i),2) = nNode

Gambar 3.7 Pseudocode Inisialisasi elemen dengan metode Level Set (3)

<i>Pseudocode 3.6</i> Inisialisasi elemen dengan metode Level Set	
Masukan	iter, CONNEC, DOMAIN, INC, NODES, XYZ, CRACK
Keluaran	Omega, PHI, PSI, ZETA
<pre> 113. nNode = nNode+1; 114. end 115. // Number the crack tip nodes 116. for i = 1:length(ctNodes) 117. NODES(ctNodes(i),4) = nNode; 118. NODES(ctNodes(i),6) = nNode+1; 119. NODES(ctNodes(i),8) = nNode+2; 120. NODES(ctNodes(i),10) = nNode+3; 121. nNode = nNode+4; 122. end 123. // Number the bimaterial crack tip nodes 124. for i = 1:length(bmNodes) 125. NODES(bmNodes(i),4) = nNode; 126. NODES(bmNodes(i),6) = nNode+1; 127. NODES(bmNodes(i),8) = nNode+2; 128. NODES(bmNodes(i),10) = nNode+3; 129. NODES(bmNodes(i),12) = nNode+4; 130. NODES(bmNodes(i),14) = nNode+5; 131. NODES(bmNodes(i),16) = nNode+6; 132. NODES(bmNodes(i),18) = nNode+7; 133. NODES(bmNodes(i),20) = nNode+8; 134. NODES(bmNodes(i),22) = nNode+9; 135. NODES(bmNodes(i),24) = nNode+10; 136. NODES(bmNodes(i),26) = nNode+11; 137. nNode = nNode+12; 138. end 139. else 140. omega ← [] 141. end </pre>	

Gambar 3.8 Pseudocode Inisialisasi elemen dengan metode Level Set (4)

Fungsi define PSI yang disebutkan pada Gambar 3.7 baris ke 79 dan 84 merupakan proses yang dieksekusi untuk mendapatkan nilai PSI. Fungsi ini dijabarkan dalam bentuk *pseudocode* pada Gambar 3.9 dan Gambar 3.10

<i>Pseudocode 3.7.</i> Define PSI
<pre> 1. dist ← zeros(nPt-1,1) 2. sine ← dist 3. for iSeg = 1 to nPt-1 4. x1 ← CRACK(iSeg,1) 5. x2 ← CRACK(iSeg+1,1) 6. y1 ← CRACK(iSeg,2) 7. y2 ← CRACK(iSeg+1,2) 8. m ← (y2-y1)/(x2-x1) </pre>

Gambar 3.9 Pseudocode fungsi untuk mendefinisikan PSI (1)

<i>Pseudocode 3.8. Define PSI</i>	
9.	if isinf(m) = 1
10.	$m \leftarrow 1e6$
11.	end
12.	$b \leftarrow y1 - m * x1$
13.	$xo \leftarrow (m * y + x - m * b) / (m^2 + 1)$
14.	$yo \leftarrow (m^2 * y + m * x + b) / (m^2 + 1)$
15.	if iSeg ~= 1
16.	if xo < x1
17.	$xo \leftarrow x1$
18.	$yo \leftarrow y1$
19.	end
20.	end if iSeg ~= (nPt-1)
21.	if xo > x2
22.	$xo \leftarrow x2$
23.	$yo \leftarrow y2$
24.	end
25.	end
26.	$dist(iSeg) \leftarrow \sqrt{(xo - x)^2 + (yo - y)^2}$
27.	$sine(iSeg) \leftarrow \text{sign}(x2 - x1) * \text{sign}(y - yo);$
28.	end
29.	$dMin \leftarrow \min(\text{abs}(dist))$
30.	$ind \leftarrow \text{find}(\text{abs}(dist) = dMin)$
31.	if length(ind) = 2
32.	$ind(2) \leftarrow []$
33.	end
34.	$psi \leftarrow dist(ind) * sine(ind)$
35.	if psi = 0
36.	$psi = 1e-6$
37.	end

Gambar 3.10 Pseudocode fungsi untuk mendefinisikan PSI (2)

3.2.5. Fungsi calcDOF

Fungsi ini merupakan fungsi yang digunakan untuk menghitung *degree of freedom* total yang mencakup *degree of freedom* dari Finite Element tradisional maupun pengayaan. Setelah terbentuk level set dengan fungsi sebelumnya, didapatkan elemen-elemen pengayaan. Hal ini yang membuat *degree of freedom* tidak dapat diinisialisasi dengan uniform secara keseluruhan. Oleh karena itu, *degree of freedom* untuk setiap elemen pengayaan dieksekusi sesuai dengan fungsi pengayaannya masing-masing. Fungsi calcDOF ini dinyatakan dalam *pseudocode* seperti pada Gambar 3.11. Perhitungan *degree of freedom* masing-masing elemen pengayaan tampak pada baris 5, 8, dan 11. Dan baris 13 dieksekusi untuk mendapatkan *degree of freedom* secara global.

Pseudocode 3.9. Menghitung total degree of freedom	
Masukan	NODES
Keluaran	DOF, DISP
<pre> 1. heavDOF ← 2*nonzero(NODES(:,2)) 2. ctipDOF ← 2*nonzero(NODES(:,4)) 3. bmctDOF ← 2*nonzero(NODES(:,12)) 4. if heavDOF > 0 5. heavNodes 6. end 7. if ctipDOF > 0 8. ctipNodes 9. end 10. if bmctDOF > 0 11. bimactipNodes 12. end 13. DOF ← 2*max(max(NODES)) 14. DISP ← sparse(DOF,1) </pre>	

Gambar 3.11. Pseudocode untuk menghitung Degree of Freedom

3.2.6. Fungsi enrElem

Fungsi ini merupakan fungsi yang digunakan untuk membentuk elemen pengayaan. Pada proses sebelumnya didapatkan informasi titik-titik mana saja yang mendapatkan pengayaan baik sebagai titik ujung sobekan maupun titik Heaviside. Dari informasi proses-proses sebelumnya tersebut dibentuk elemen sesuai dengan titik-titik pengayaan yang berkesinambungan. Fungsi ini dieksekusi sesuai dengan *pseudocode* seperti pada Gambar 3.12 dan Gambar 3.13

Pseudocode 3.10 Membentuk elemen pengayaan	
Masukan	i, pHDOF, NODES, CONNEC
Keluaran	enrElem, IElem
<pre> 1. I ← find NODES(:,4)>0 2. c1 ← intersect(I,CONNEC(:,2)') 3. c2 ← intersect(I,CONNEC(:,3)') 4. c3 ← intersect(I,CONNEC(:,4)') 5. c4 ← intersect(I,CONNEC(:,5)') 6. CTElem ← unique(vertcat(c1,c2,c3,c4)) 7. if i = 1 8. I ← find NODES(:,2)>0 9. c1 ← intersect(I,CONNEC(:,2)') 10. c2 ← intersect(I,CONNEC(:,3)') 11. c3 ← intersect(I,CONNEC(:,4)') 12. c4 ← intersect(I,CONNEC(:,5)') 13. HElem ← unique(vertcat(c1,c2,c3,c4)) 14. I ← find NODES(:,30)>0 </pre>	

Gambar 3.12.Pseudocode untuk membentuk elemen pengayaan (1)

Pseudocode 3.11 Membentuk elemen pengayaan	
Masukan	i, pHDOF, NODES, CONNEC
Keluaran	enrElem, IElem
<pre> 15. c1 ← intersect(I, CONNEC(:,2)') 16. c2 ← intersect(I, CONNEC(:,3)') 17. c3 ← intersect(I, CONNEC(:,4)') 18. c4 ← intersect(I, CONNEC(:,5)') 19. IElem ← unique(vertcat(c1,c2,c3,c4)) 20. else 21. I ← find NODES(:,2) > pHDOF/2 22. c1 ← intersect(I, CONNEC(:,2)') 23. c2 ← intersect(I, CONNEC(:,3)') 24. c3 ← intersect(I, CONNEC(:,4)') 25. c4 ← intersect(I, CONNEC(:,5)') 26. HElem ← unique(vertcat(c1,c2,c3,c4)) 27. IElem ← [] 28. end 29. enrElem = unique(vertcat(CTElem, HElem)) </pre>	

Gambar 3.13 Pseudocode untuk membentuk elemen pengayaan (2)

3.2.7. Fungsi stiffnessMatrix dan updateStiffness

Fungsi yang menghitung matrik kekakuan ini dibagi menjadi 2 yaitu stiffnessMatrix yang digunakan untuk menghitung kekakuan dasar pada iterasi pertama dan updateStiffness yang digunakan untuk menghitung kekakuan pada iterasi-iterasi berikutnya. Perbedaan fungsi updateStiffness adalah fungsi ini tidak menghitung matrik kekakuan elemen inklusi karena elemen inklusi tidak mengalami perubahan terhadap fungsi waktu. Fungsi ini dikerjakan sesuai dengan pseudocode seperti pada Gambar 3.14, Gambar 3.15, Gambar 3.16, Gambar 3.17, Gambar 3.18, dan Gambar 3.19

Pseudocode 3.12 Membentuk matrik kekakuan	
Masukan	Omega, DOF, iter, enrElem, IElem, CONNEC, CRACK, DOMAIN, MAT, NODES, PHI, PSI, XYZ, ZETA
Keluaran	globalK
<pre> 1. nXElem ← DOMAIN(1) 2. nYElem ← DOMAIN(2) 3. lXElem ← DOMAIN(3) 4. Em ← MAT(1) 5. Vm ← MAT(2) 6. Ef ← MAT(3) 7. Vf ← MAT(4) </pre>	

Gambar 3.14 Pseudocode untuk membangun matrik kekakuan (1)

Pseudocode 3.13 Membentuk matrik kekakuan	
Masukan	Omega, DOF, iter, enrElem, IElem, CONNEC, CRACK, DOMAIN, MAT, NODES, PHI, PSI, XYZ, ZETA
Keluaran	globalK
<pre> 8. plane ← MAT(5) 9. nCT ← size(PHI) 10. globalK ← sparse matrix size globalDOF 11. if plane = 1 12. build initial plane stress stiffness matrix 13. Km = build initial plane stress stiffness fiber matrix 14. elseif plane = 2 15. build initial plane strain stiffness matrix 16. Kf = build initial plane strain stiffness fiber matrix 17. end 18. define crack point coordinate 19. for iElem = 1 to nYElem*nXElem 20. NN ← current elements nodal data 21. CN ← chi level set values 22. CTN ← number of crack tip node enrichment 23. HEN ← number of Heaviside node enrichment 24. IEN ← number of inclusion node enrichment 25. NEN ← number of enriched nodes 26. lokalK ← 0 27. local = traditional index 28. iLoc ← 9 29. if NEN = 0 Set stiffness matrix for traditional nodes 30. //Set stiffness for enriched nodes 31. if NEN > 0 32. define nodes coordinates for current element 33. if NEN = 4 //fully enriched element 34. define crack level set values 35. define inclusion level set values 36. [gp, gw, J] = subDomain(enriched nodes) 37. for i = 1 to length of Gauss Point 38. defines Gauss elements 39. end 40. elseif min(CN)<0 //partially enriched element 41. define crack level set values 42. define inclusion level set values 43. [gp, gw, J] = subDomain(enriched nodes) 44. for i = 1 to length of Gauss Point 45. defines Gauss elements 46. end 47. else </pre>	

Gambar 3.15 Pseudocode untuk membangun matrik kekakuan (2)

Pseudocode 3.14 Membentuk matrik kekakuan	
Masukan	Omega, DOF, iter, enrElem, IElem, CONNEC, CRACK, DOMAIN, MAT, NODES, PHI, PSI, XYZ, ZETA
Keluaran	globalK
<pre> 48. [gp,gw]=gauss(6,'QUAD') 49. J = []; 50. end 51. for i = 1 to length of Gauss Point 52. define Gauss points 53. define Gauss weight 54. define Jacobian subdomain 55. define determinant of Jacobian 56. define shape function 57. calculate shape derivative with respect to x 58. calculate shape derivative with respect to y 59. define nodal x coordinate of element 60. define nodal y coordinate of element 61. define global x for current gauss point 62. define global y for current gauss point 63. define material level set at current gauss point 64. Benr ← [] 65. Bu ← [[Nx(1) 0 Nx(2) 0 Nx(3) 0 Nx(4) 0 0 Ny(1) 0 Ny(2) 0 Ny(3) 0 Ny(4) Ny(1) Nx(1) Ny(2) Nx(2) Ny(3) Nx(3) Ny(4) Nx(4)]; 66. index ← 1//define local stiffness based on enriched elements 67. for iN = 1 to 4 68. if NN(iN,2) ~= 0 69. Define PSI level set value at current gauss point 70. Hgp ← sign(psi) 71. Hi ← NN(iN,3) 72. H ← Hgp-Hi 73. Ba ← [Nx(iN)*H 0 ; 0 Ny(iN)*H; Ny(iN)*H Nx(iN)*H]; 74. Benr(:,index to index+1) = Ba 75. index = index+2 76. if I = length of Gauss Point 77. local(iLoc to iLoc+1) ← [2*NN(iN,2)-1 2*NN(iN,2)] 78. iLoc ← iLoc+2 79. end 80. else if NN(iN,4) ~= 0 81. if nCt = 1 82. define distance crack tip to gauss point 83. CCS ←[cos(omega) sin(omega) </pre>	

Gambar 3.16 Pseudocode untuk membangun matrik kekakuan (3)

Pseudocode 3.15 Membentuk matrik kekakuan	
Masukan	Omega, DOF, iter, enrElem, IElem, CONNEC, CRACK, DOMAIN, MAT, NODES, PHI, PSI, XYZ, ZETA
Keluaran	globalK
84.	$[-\sin(\omega) \cos(\omega)]$
85.	$XYLoc \leftarrow CCS * distance$
86.	$r \leftarrow \sqrt{XYLoc(1)^2 + XYLoc(2)^2}$
87.	if $r < 0.001 * lxElem$
88.	$r \leftarrow 0.05 * lxElem$
89.	end
90.	$\theta \leftarrow \text{atan}(XYLoc)$
91.	else if $nCt = 2$
92.	define distance crack node point 1 to gauss point
93.	define distance crack node point 2 to gauss point
94.	$CCS \leftarrow [\cos(\omega(1)) \sin(\omega(1))$
95.	$\quad -\sin(\omega(1)) \cos(\omega(1))]$
96.	$XYLoc1 \leftarrow CCS * distance1$
97.	$CCS \leftarrow [\cos(\omega(2)) \sin(\omega(2))$
98.	$\quad -\sin(\omega(2)) \cos(\omega(2))]$
99.	$XYLoc2 \leftarrow CCS * distance2$
100.	$r1 \leftarrow \sqrt{XYLoc1(1)^2 + XYLoc1(2)^2}$
101.	$r2 \leftarrow \sqrt{XYLoc2(1)^2 + XYLoc2(2)^2}$
102.	if $r1 > r2$
103.	$r \leftarrow r2$
104.	$\theta \leftarrow \text{atan}(XYLoc2)$
105.	else if $r2 > r1$
106.	$r \leftarrow r1$
107.	$\theta \leftarrow \text{atan}(XYLoc1)$
108.	end
109.	if $r < 0.001 * lxElem$
110.	$r \leftarrow 0.05 * lxElem$
111.	end
112.	end
113.	end
114.	if $NN(iN, 12) = 0$
115.	//define crack tip enrichment value
116.	$a1gp = \sqrt{r} * \sin(\theta/2);$
117.	$a2gp = \sqrt{r} * \cos(\theta/2);$
118.	$a3gp = \sqrt{r} * \sin(\theta) * \sin(\theta/2);$
119.	$a4gp = \sqrt{r} * \sin(\theta) * \cos(\theta/2);$
120.	//define shifted alpha enrichment value
121.	$a1 = a1gp - NN(iN, 5);$
122.	$a2 = a2gp - NN(iN, 7);$
123.	$a3 = a3gp - NN(iN, 9);$
124.	$a4 = a4gp - NN(iN, 11);$
125.	derivate crack tip enrichment function with respect to x

Gambar 3.17 Pseudocode untuk membangun matrik kekakuan (4)

Pseudocode 3.16 Membentuk matrik kekakuan	
Masukan	Omega, DOF, iter, enrElem, IElem, CONNEC, CRACK, DOMAIN, MAT, NODES, PHI, PSI, XYZ, ZETA
Keluaran	globalK
126.	derivate crack tip enrichment function with respect to y
127.	Bb = build local stiffness matrix Benr(:,index to index+7) = Bb
128.	Index \leftarrow index+8
129.	if i = length of Gauss Point
130.	local(iLoc:(iLoc+7)) = [2*NN(iN,4)-1 2*NN(iN,4) 2*NN(iN,6)-1 2*NN(iN,6) 2*NN(iN,8)-1 2*NN(iN,8) 2*NN(iN,10)-1 2*NN(iN,10)];
131.	iLoc = iLoc+8;
132.	end
133.	else
134.	Define constant for bimaterial crack tip
135.	Define bimaterial crack tip value
136.	Define shifted alpha enrichment value
137.	Derivative of bimaterial crack tip enrichment functions with respect to crack tip coordinate system x coordinate
138.	Derivative of bimaterial crack tip enrichment functions with respect to crack tip coordinate system y coordinate
139.	Derivative of bimaterial crack tip enrichment functions with respect to X
140.	Derivative of bimaterial crack tip enrichment functions with respect to Y
141.	Bb = build local stiffness matrix
142.	Benr(:,index:(index+23)) \leftarrow Bb;
143.	Index \leftarrow index+24
144.	if i == length(gp)
145.	for iB = 4:2:26
146.	local(iLoc:(iLoc+1)) = [2*NN(iN,iB)-1 2*NN(iN,iB)];
147.	iLoc = iLoc+2;
148.	end
149.	end
150.	end
151.	if NN(iN,30) ~= 0
152.	define inclusion enrichment constant
153.	Ba = [Nx(iN)*E+N(iN)*Ex 0; 0 Ny(iN)*E+N(iN)*Ey; Ny(iN)*E+N(iN)*Ey Nx(iN)*E+N(iN)*Ex]
154.	Benr(:, index to index+1) \leftarrow Ba
155.	Index \leftarrow index+2
156.	If I = 1 to length of Gauss Point

Gambar 3.18. Pseudocode untuk membangun matrik kekakuan (5)

Pseudocode 3.17 Membentuk matrik kekakuan	
Masukan	Omega, DOF, iter, enrElem, IElem, CONNEC, CRACK, DOMAIN, MAT, NODES, PHI, PSI, XYZ, ZETA
Keluaran	globalK
<pre> 157. local(iLoc:(iLoc+1)) ← [2*NN(iN,30)-1 2*NN(iN,30)] 158. iLoc ← iLoc+2 159. end 160. end 161. B ← [Bu Benr]; 162. localK ← localK + W*B'*C*B*detJ; 163. end 164. end 165. if length(localK) == 8 % Unenriched element 166. for row = 1:8 167. for col = 1:8 168. nIndexT = nIndexT+1; 169. allRowT(nIndexT) = local(row); 170. allColT(nIndexT) = local(col); 171. allValT(nIndexT) = localK(row,col); 172. end 173. end 174. else 175. //Assemble the global stiffness 176. globalK(local,local) = globalK(local,local) + localK; 177. end 178. end 179. globalK = globalK + sparse(allRowT,allColT,allValT,globalDOF,globalDOF); </pre>	

Gambar 3.19 Pseudocode untuk membangun matrik kekakuan (6)

3.2.8. Fungsi forceVector

Fungsi forceVector digunakan untuk mengaplikasikan besaran gaya yang telah dimasukkan oleh pengguna ke dalam bidang domain. Besaran gaya ini mempengaruhi panjang dari sobekan baru yang terbentuk. Fungsi ini dieksekusi sesuai *pseudocode* pada Gambar 3.20 dan Gambar 3.21.

Pseudocode 3.18 Fungsi untuk membentuk forceVector global	
Masukan	DOF, I, DOMAIN, MAT, FORCE, CONNEC
Keluaran	globalF
<pre> 1. nXElem ← DOMAIN(1) 2. nYElem ← DOMAIN(2) 3. le ← DOMAIN(3) 4. globalF ← sparse(DOF,1) 5. m ← size(FORCE,1); 6. if MAT(5) = 1 7. h ← MAT(6) </pre>	

Gambar 3.20. Pseudocode fungsi forceVector(1)

Pseudocode 3.19 Fungsi untuk membentuk forceVector global	
Masukan	DOF, I, DOMAIN, MAT, FORCE, CONNEC
Keluaran	globalF
<pre> 8. else 9. h ← 1 10. end 11. for iForce = 1 to m 12. edge ← FORCE(iForce,1) 13. xForce ← FORCE(iForce,2) 14. yForce ← FORCE(iForce,3) 15. end 16. node ← CONNEC((nYElem-1)*(nXElem)+1,5) 17. for i = 1 to nXElem+1 18. if i = 1 i = nXElem+1 19. globalF(2*node-1,1) = le*xForce/2 20. globalF(2*node,1) = le*yForce/2 21. else 22. globalF(2*node-1,1) = le*xForce 23. globalF(2*node,1) = le*yForce 24. end 25. node = node+1 26. end </pre>	

Gambar 3.21 Pseudocode fungsi forceVector (2)

3.2.9. Fungsi boundaryCond

Fungsi boundaryCond ini dijalankan untuk mendefinisikan *degree of freedom* dari *boundary condition* atau wilayah-wilayah tepi dari domain yang harus diterapkan. Proses eksekusi fungsi ini dilakukan sesuai dengan *pseudocode* Gambar 3.22.

Pseudocode 3.20 Mendefinisikan boundary condition	
Masukan	DOMAIN, DOF
Keluaran	freeDOF
<pre> nXElem ← DOMAIN(1) nYElem ← DOMAIN(2) fixedDOF ← [2*(nXElem+1)-1 2*(nXElem+1) 2*(nXElem+1)*(nYElem+1)-1] freeDOF ← setdiff(1 to DOF, fixedDOF) </pre>	

Gambar 3.22 Pseudocode untuk mendefinisikan boundary condition

3.2.10. Fungsi JIntegral

Fungsi ini diimplementasikan untuk menghitung mode gabungan yang berkaitan dengan faktor intensitas tegangan untuk domain yang sobek. Fungsi ini merupakan representasi kode dari bab 2.3. Fungsi ini dijalankan sesuai dengan *pseudocode* pada Gambar 3.23, Gambar 3.24, dan Gambar 3.25

Pseudocode 3.21 Penghitungan factor intensitas tegangan dengan J-Integral	
Masukan	Omega, DISP, CONNEC, CRACK, DOMAIN, MAT, NODES, PSI, XYZ, ZETA
Keluaran	KI, KII
<pre> 1. nElem ← DOMAIN(1)*DOMAIN(2); 2. nNode ← (DOMAIN(1)+1)*(DOMAIN(2)+1); 3. lXElem ← DOMAIN(3); 4. Em ← MAT(1); 5. vm ← MAT(2); 6. Ef ← MAT(3); 7. vf ← MAT(4); 8. plane ← MAT(5); 9. Gm ← Em/2/(1+vm); 10. Gf ← Ef/2/(1+vf); 11. Create elastic constant matrix 12. nCT = length(omega); 13. for iJ = 1 to nCT 14. determine number of data points defining crack 15. CCS = [cos(omega(iJ)) sin(omega(iJ)); -sin(omega(iJ)) cos(omega(iJ))] 16. area ← lXElem*lXElem 17. Define components to be used in J-Integral 18. Determine elements in J-Integral and assign nodal q values 19. Jdomain = elements within search radius 20. For iElem = 1 to length of Jdomain 21. Define points for current elements 22. Define Gauss point and gauss weight 23. Define quadrilateral shape function and derivative 24. strain ← B*Xe 25. stress ← C*strain 26. if nnz(NODES(:,12))==0 27. calculate crack auxiliary fields constant 28. for mode = 1 to 2 29. if mode = 1 30. calculate auxiliary field components from Mode I 31. else if mode = 2 32. calculate auxiliary field components from Mode I 33. end 34. calculate I1 35. calculate I2 </pre>	

Gambar 3.23 Pseudocode fungsi penghitungan faktor intensitas tegangan dengan J-Integral (1)

Pseudocode 3.22 Penghitungan factor intensitas tegangan dengan J-Integral	
Masukan	Omega, DISP, CONNEC, CRACK, DOMAIN, MAT, NODES, PSI, XYZ, ZETA
Keluaran	KI, KII
<pre> 36. StrainEnergy = 0; 37. for j = 1 to 2 38. for k = 1 to 2 39. StrainEnergy = StrainEnergy+CalStress(j,k)*AuxStrain(j,k); 40. end 41. end 42. I(mode,1) ← I(mode,1)+(I1+I2- StrainEnergy*GradQ(1))*detJ*W; 43. end 44. else 45. define material constant from enriched elements 46. calculate crack auxiliary constant 47. for mode = 1 to 2 48. if mode = 1 49. calculate auxiliary field components from Mode I 50. else if mode = 2 51. calculate auxiliary field components from Mode I 52. end 53. calculate I1 54. calculate I2 55. StrainEnergy = 0; 56. for j = 1 to 2 57. for k = 1 to 2 58. StrainEnergy = StrainEnergy+CalStress(j,k)*AuxStrain(j,k); 59. End 60. end 61. I(mode,1) ← I(mode,1)+(I1+I2- StrainEnergy*GradQ(1))*detJ*W; 62. end 63. end 64. end 65. if isempty(setdiff(NODES(:,12)),0) = 1 66. if plane == 1 67. Eeff = Em; 68. elseif plane == 2 69. Eeff = Em/(1-vm^2); 70. end </pre>	

Gambar 3.24 Pseudocode fungsi penghitungan faktor intensitas tegangan dengan J-Integral (2)

Pseudocode 3.23 Penghitungan factor intensitas tegangan dengan J-Integral	
Masukan	Omega, DISP, CONNEC, CRACK, DOMAIN, MAT, NODES, PSI, XYZ, ZETA
Keluaran	KI, KII
<pre> 71. Kcalc = I*Eeff/2; 72. KI(iJ) = Kcalc(1); 73. KII(iJ) = Kcalc(2); 74. 75. else 76. if plane == 1 77. Emeff = Em; 78. Efeff = Ef; 79. elseif plane == 2 80. Emeff = Em/(1-vm^2); 81. Efeff = Ef/(1-vf^2); 82. end 83. 84. Eeff = 2*Emeff*Efeff/(Emeff+Efeff); 85. Kcalc = I*Eeff/2*cosh(e*pi)*cosh(e*pi); 86. KI(iJ) = Kcalc(1); 87. KII(iJ) = Kcalc(2); 88. end 89. end </pre>	

Gambar 3.25 Pseudocode fungsi penghitungan faktor intensitas tegangan dengan J-Integral (3)

3.2.11. Fungsi growCrack

Fungsi ini akan mempengaruhi besar sobekan sesuai dengan kriteria yang dimasukkan oleh pengguna. Fungsi ini dijabarkan dengan *pseudocode* pada Gambar 3.26 dan Gambar 3.27. Selain ditentukan oleh pengguna, sudut retakan dipengaruhi pula oleh maximum circumferential stress yang dieksekusi pada baris ke 7.

Pseudocode 3.24. Pseudocode fungsi growCrack	
Masukan	KI, KII, omega, CRACK, MAT, GROW
Keluaran	exit
<pre> 1. Kc ← MAT(7) 2. Exit ← 'NO' 3. nCT ← length(omega) 4. for iCT = 1 to nCT 5. K1 ← KI(iCT) 6. K2 ← KII(iCT) 7. thetaC ← dirMaximumCircumferentialStress(K1,K2,omega(iCT)) </pre>	

Gambar 3.26. Pseudocode fungsi growCrack (1)

Pseudocode 3.25. <i>Pseudocode</i> fungsi growCrack	
Masukan	KI, KII, omega, CRACK, MAT, GROW
Keluaran	exit
<pre> 8. dK1 ← K1 9. dK2 ← K2 10. dK ← eqSIFYan(dK1,dK2, thetaC) 11. if dK < Kc 12. nSeg ← size(CRACK,1) 13. da ← GROW(2) 14. if da > 0 15. dx ← da*cos(thetaC) 16. dy ← da*sin(thetaC) 17. else 18. dx ← 0 19. dy ← 0 20. end 21. if iCT = 1 22. CRACK(nSeg+1,:) = [CRACK(nSeg,1)+dx,CRACK(nSeg,2)+dy]; 23. else if iCT = 2 24. newCR = zeros(nSeg+1,2); 25. newCR(2:nSeg+1,:) = CRACK; 26. newCR(1,:) = [CRACK(1,1)+dx,CRACK(1,2)+dy]; 27. CRACK = newCR; 28. end 29. else 30. exit = 'YES' 31. end 32. end </pre>	

Gambar 3.27. Pseudocode fungsi growCrack (2)

BAB 4

UJI COBA DAN PEMBAHASAN

Bab ini menjelaskan uji coba yang dilakukan pada model yang dibangun beserta hasil dan analisis uji coba. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba, beserta analisis hasil dari setiap skenario pengujian.

4.1. Lingkungan Uji Coba

Lingkungan uji coba menjelaskan spesifikasi dari perangkat keras dan perangkat lunak yang digunakan untuk menguji model yang telah dibangun. Model yang diuji coba adalah model sobekan. Penjelasan dari perangkat keras dan perangkat lunak yang digunakan yaitu sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™ i3 CPU 540 @ 3.07GHz(4 CPUs),~3,1GHz
 - b. Memori (RAM): 4096MB RAM
 - c. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak:
 - a. Sistem operasi: Windows 8.1 Po 64-bit(6.3, Build 9600)
 - b. Perangkat pengembang: MATLAB R2016a

4.2. Skenario Uji Coba

Uji coba dilakukan dengan dua skenario utama yaitu model yang diuji coba tanpa titik inklusi dan model yang diuji coba dengan titik inklusi. Diharapkan bahwa dari kedua uji coba utama ini didapatkan hasil yang signifikan sehingga dapat diamati variabel-variabel yang mempengaruhi model dalam membentuk objek yang sesuai dengan tujuan penelitian.

Untuk percobaan diterapkan masukan yang telah dijelaskan pada Bab 3.2.1. Secara singkat, masukan tersebut dijabarkan dalam Tabel 4.1. Masukan ini yang nantinya akan diaplikasikan ke setiap percobaan dengan nilai yang bervariasi.

Pada dua uji coba utama akan dilakukan beberapa variasi variabel. Diantaranya adalah variasi nilai variabel modulus Young dan rasio Poisson. Hasil yang diamati adalah grid dengan Finite Element yang dihasilkan, *mesh* dalam bentuk deformasi, dan waktu komputasi.

Tabel 4.1. Masukan model sobekan

Variabel	Keterangan
Height	Tinggi objek (satuan panjang)
Width	Lebar objek (satuan panjang)
LElem	Ukuran tiap elemen
DOMAIN	[ukuran x, ukuran y, ukuran elemen x, ukuran elemen y]
MAT	[Young modulus domain, rasio Poisson domain, Young modulus inklusi, rasio Poisson inklusi, status sress-strain, ukuran plane stress, faktor intensitas tegangan kritis pada domain];
CRACK	[x1, y1, x2, y2]
INC	[inklusi]
GROW	[iterasi, crack growth increment];
FORCE	[uniaxial tension, besaran gaya pada x, besaran gaya pada y]

Pada Tabel 4.1, terdapat tinggi dan lebar objek yang diterapkan menggunakan satuan panjang atau *unit length*. Setiap satu satuan panjang akan dibagi ke dalam piksel-piksel yang jumlahnya ditentukan oleh ukuran LElem. Apabila diberikan panjang 1 dengan LElem 1/20 atau 0.05, maka dalam satu satuan akan terdapat 20 piksel.

Pada Tabel 4.1 juga terdapat variabel modulus Young dan rasio Poisson. Nilai variasi modulus Young dan rasio Poisson ini diasumsikan berdasarkan Lampiran A dan B yang menyatakan nilai-nilai modulus Young dan rasio Poisson pada berbagai macam material.

Sesuai Material Data Book pada Lampiran A, nilai variasi modulus Young yang diasumsikan berada pada jangkauan 5 GPa-10 GPa dengan 1 GPa 1,000,000,000 Pa. Jangkauan ini ditentukan berdasarkan sifat kertas yang tidak kaku seperti logam dan tidak lentur seperti elastomer. Sedangkan nilai rasio Poisson, sesuai dengan Lampiran B diasumsikan berada pada jangkauan 0.1 hingga 0.3.

4.2.1. Uji Coba Model Sobekan Tanpa Inklusi

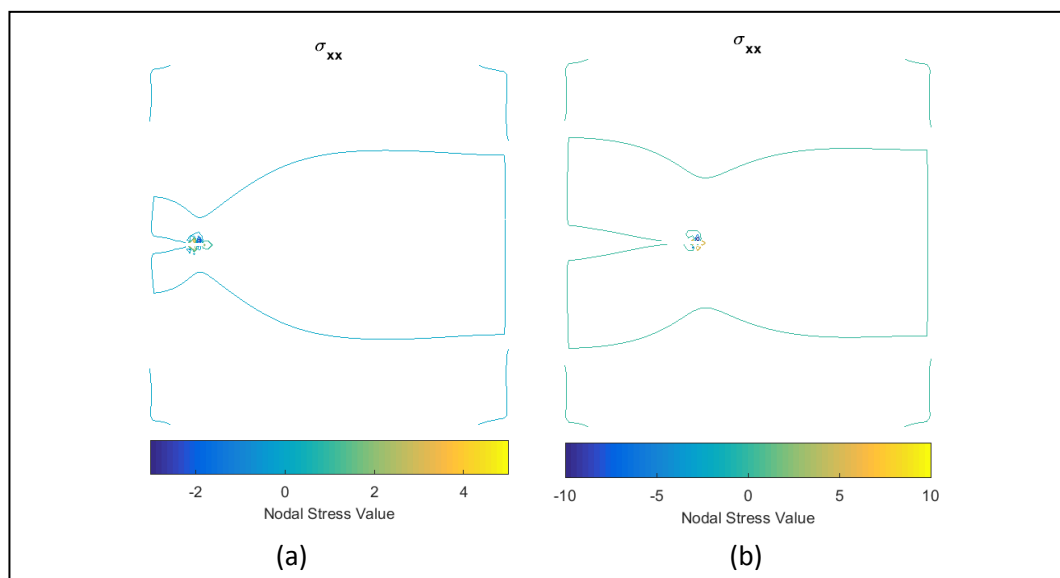
a. Percobaan Sobekan Tanpa Inklusi Variasi 1

Percobaan dilakukan dengan masukan seperti pada Tabel 4.2. Uji coba pertama ini diselesaikan dalam rata-rata waktu 23.67 detik untuk 30 kali iterasi.

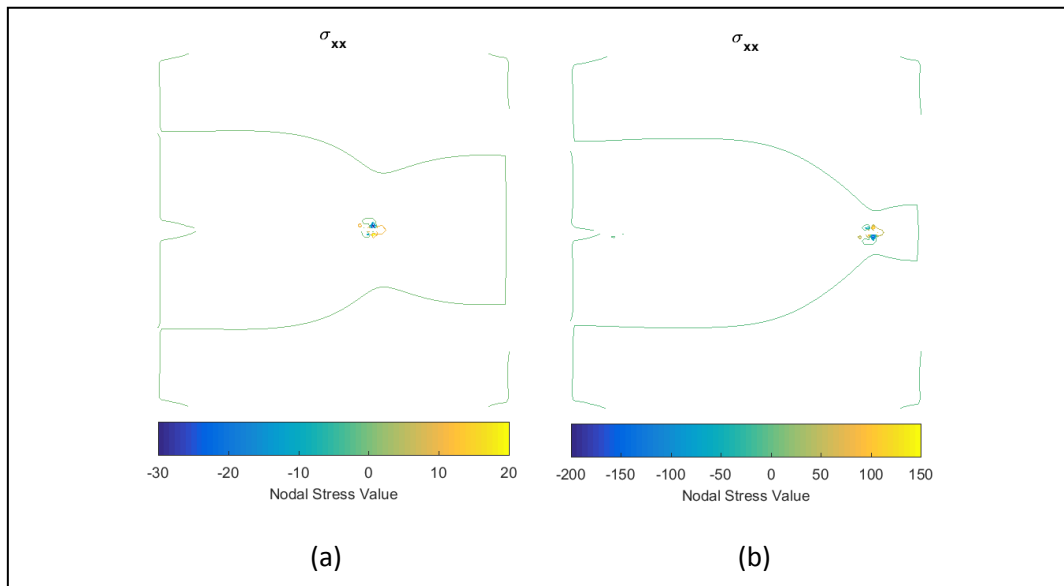
Tabel 4.2. Masukan untuk percobaan sobekan tanpa inklusi variasi 1

Variabel	Nilai
Height	4
Width	4
LElem	1/20;
DOMAIN	[Width/LElem,Height/LElem,LElem,LElem];
MAT	[10E9,0.3,10E8,0.3,2,0,Inf];
CRACK	[0 Height/2;0.5 Height/2]
INC	[]
GROW	[30 0.1];
FORCE	[1 0 1]

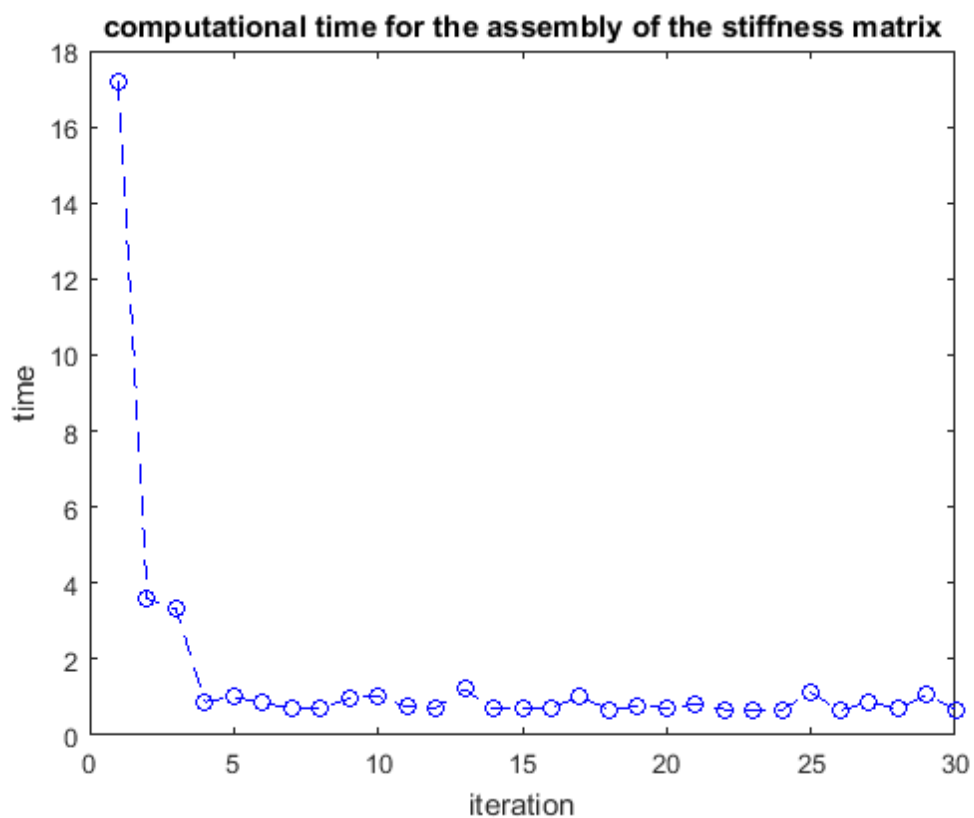
Hasil proses direpresentasikan persebaran tekanan/*stress* yang terjadi pada setiap iterasinya. Persebaran tekanan ini divisualisasikan pada Gambar 4.1 dan Gambar 4.2. Sedangkan implementasi penggabungan atau *assembly* matrik kekakuan terhadap fungsi waktu yang diimplementasikan pada model ditunjukkan pada Gambar 4.3.



**Gambar 4.1. Persebaran Stress pada proses sobekan tanpa inklusi variasi 1
(a) iterasi ke-1 (b) iterasi ke-10**



Gambar 4.2. Persebaran Stress pada proses sobekan tanpa inklusi variasi 1
(a) iterasi ke-20 (c) iterasi ke-30



Gambar 4.3 Grafik implementasi *assembly* matrik kekakuan terhadap fungsi waktu percobaan tanpa inklusi variasi 1

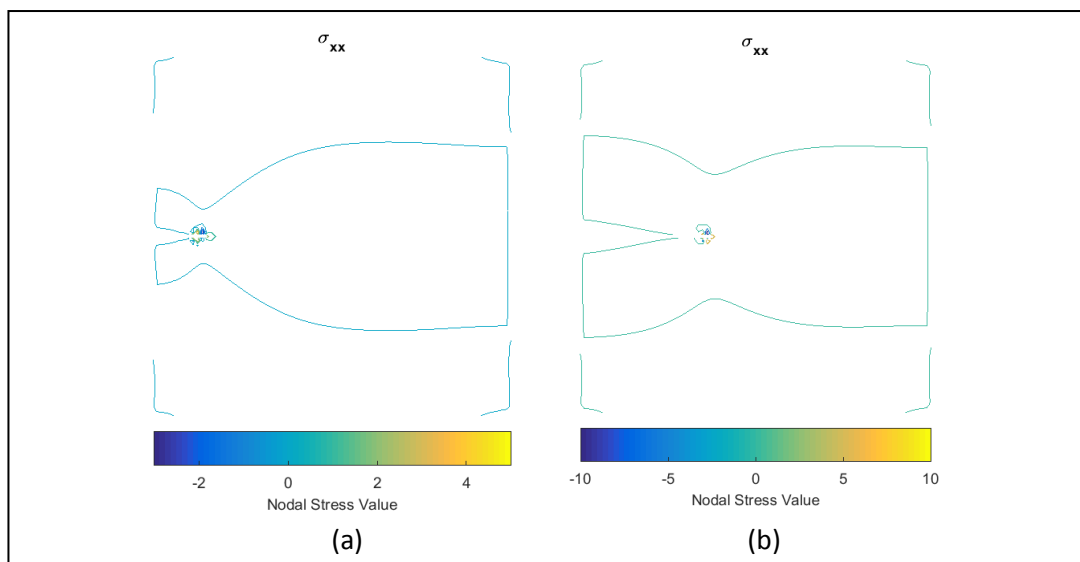
b. Percobaan Sobekan Tanpa Inklusi Variasi 2

Percobaan dilakukan dengan masukan seperti pada Tabel 4.3. Dilakukan variasi terhadap nilai modulus Young untuk domain dan inklusi. Modulus Young pada variasi ini bernilai 5E9. Uji coba pertama ini diselesaikan dalam rata-rata waktu 21.34 detik untuk 30 kali iterasi.

Tabel 4.3. Masukan untuk percobaan sobekan tanpa inklusi variasi 2

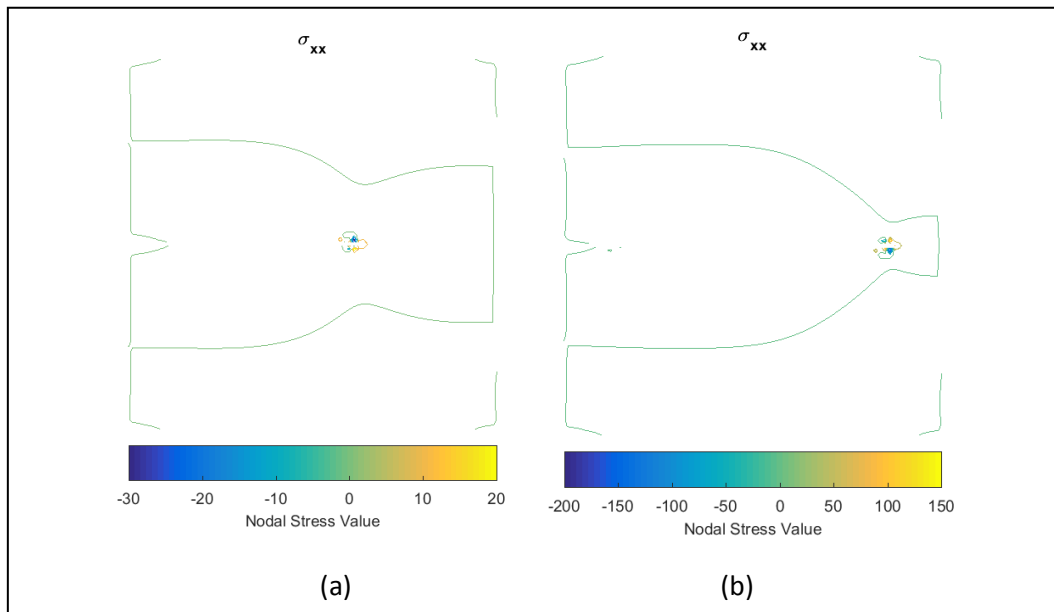
Variabel	Nilai
Height	4
Width	4
LElem	1/20;
DOMAIN	[Width/LElem,Height/LElem,LElem,LElem];
MAT	[5E9,0.3,5E8,0.3,2,0,Inf];
CRACK	[0 Height/2;0.5 Height/2]
INC	[]
GROW	[30 0.1];
FORCE	[1 0 1]

Hasil proses direpresentasikan persebaran tekanan/*stress* yang terjadi pada setiap iterasinya. Persebaran tekanan ini divisualisasikan pada Gambar 4.4 dan Gambar 4.5.



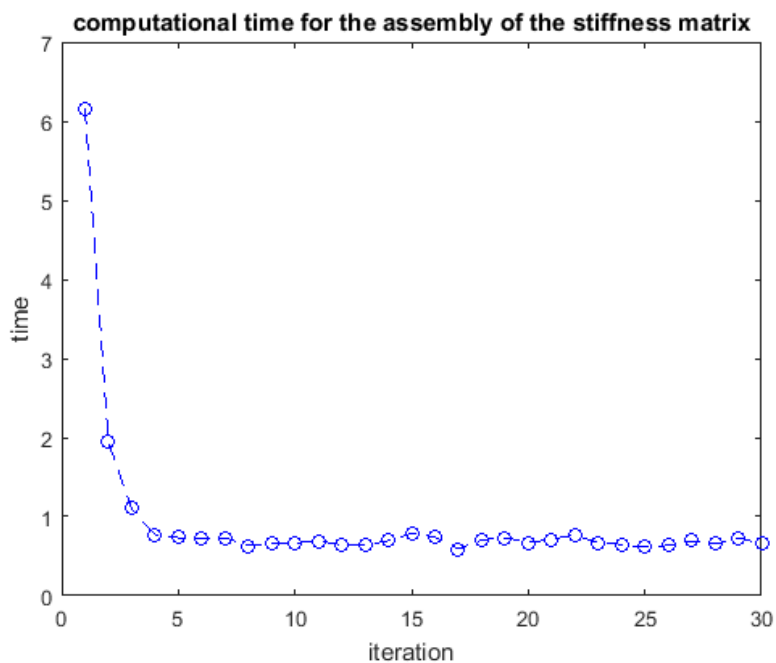
Gambar 4.4. Persebaran Stress pada proses sobekan tanpa inklusi variasi 2

(a) iterasi ke-1 (b) iterasi ke-10



Gambar 4.5 Persebaran Stress pada proses sobekan tanpa inklusi variasi 2
(a) iterasi ke-20 (b) iterasi ke-30

Sedangkan implementasi penggabungan atau *assembly* matrik kekakuan terhadap fungsi waktu yang diimplementasikan pada model ditunjukkan pada Gambar 4.6.



Gambar 4.6. Grafik implementasi assembly matrik kekakuan terhadap fungsi waktu percobaan tanpa inklusi variasi 2

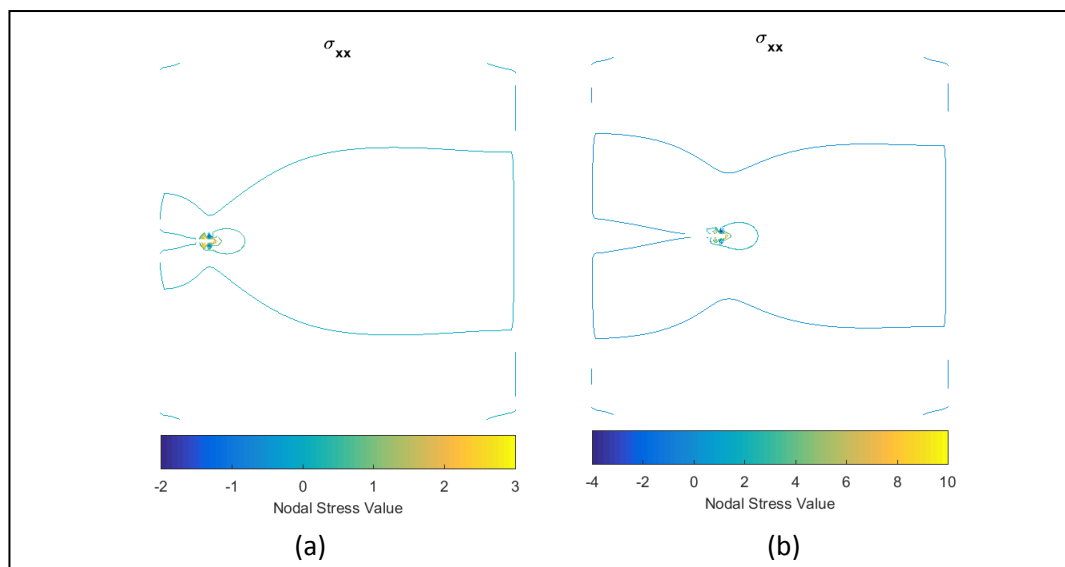
c. Percobaan Sobekan Tanpa Inklusi Variasi 3

Percobaan dilakukan dengan masukan seperti pada Tabel 4.4. Dilakukan variasi terhadap nilai rasio Poisson untuk domain dan inklusi. Rasio Poisson pada variasi ini bernilai 0.1. Uji coba pertama ini diselesaikan dalam rata-rata waktu 24.49 detik untuk 30 kali iterasi.

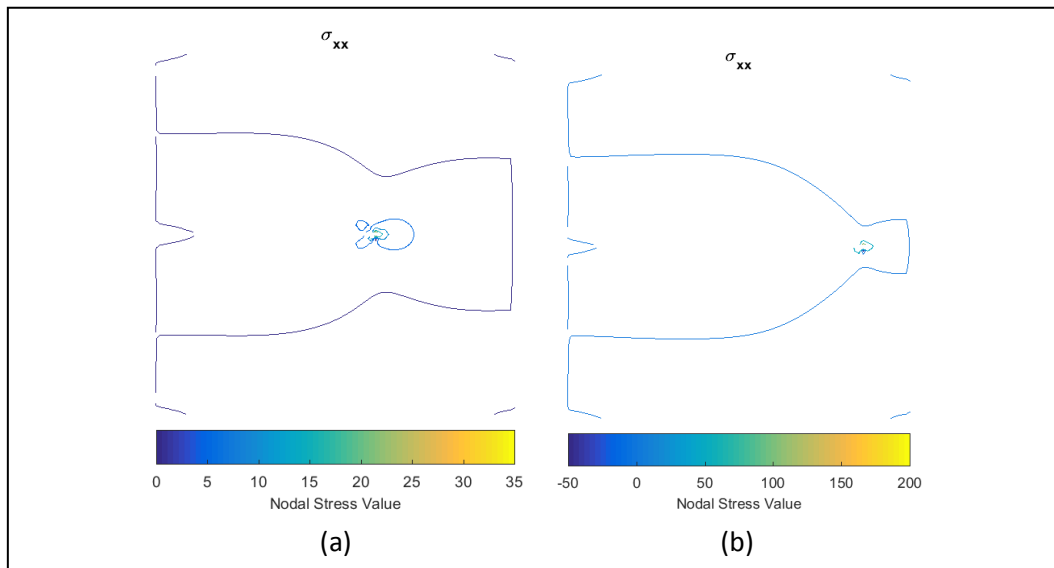
Tabel 4.4. Masukan untuk percobaan sobekan tanpa inklusi variasi 3

Variabel	Nilai
Height	4
Width	4
LElem	1/20;
DOMAIN	[Width/LElem,Height/LElem,LElem,LElem];
MAT	[10E9,0.1,10E8,0.1,2,0,Inf];
CRACK	[0 Height/2;0.5 Height/2]
INC	[]
GROW	[30 0.1];
FORCE	[1 0 1]

Hasil proses direpresentasikan persebaran tekanan/*stress* yang terjadi pada setiap iterasinya. Persebaran tekanan ini divisualisasikan pada Gambar 4.8.dan Gambar 4.8.

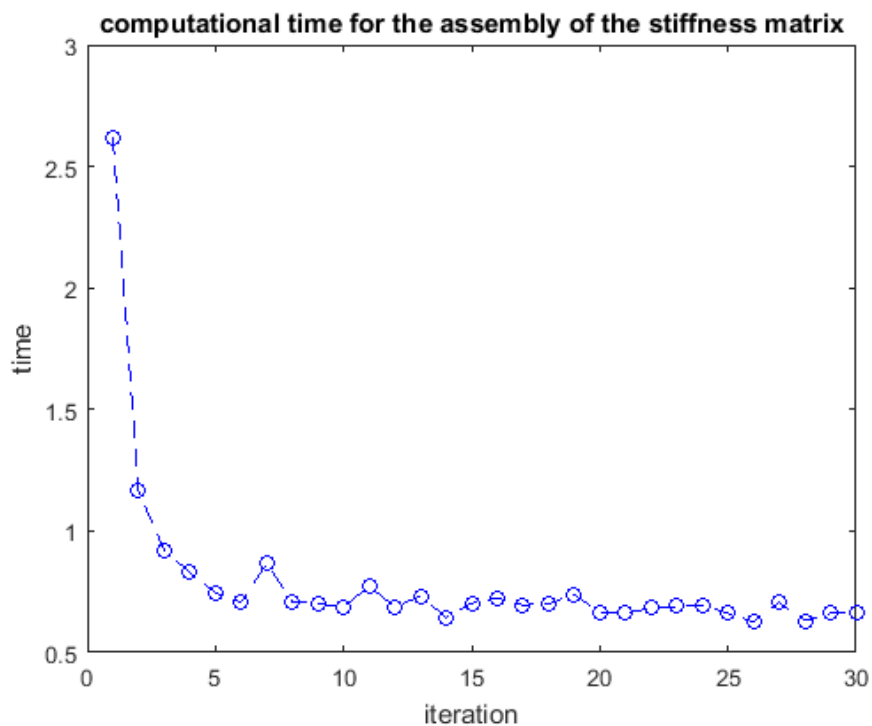


Gambar 4.7. Persebaran Stress pada proses sobekan tanpa inklusi variasi 3
(a) iterasi ke-1 (b) iterasi ke-10



Gambar 4.8. Persebaran Stress pada proses sobekan tanpa inklusi variasi 3
(a) iterasi ke-20 (b) iterasi ke-30

Sedangkan implementasi penggabungan atau *assembly* matrik kekakuan terhadap fungsi waktu yang diimplementasikan pada model untuk percobaan ini ditunjukkan pada Gambar 4.9.



Gambar 4.9. Grafik implementasi *assembly* matrik kekakuan terhadap fungsi waktu percobaan tanpa inklusi variasi 3

4.2.2. Uji Coba Model Sobekan dengan Inklusi

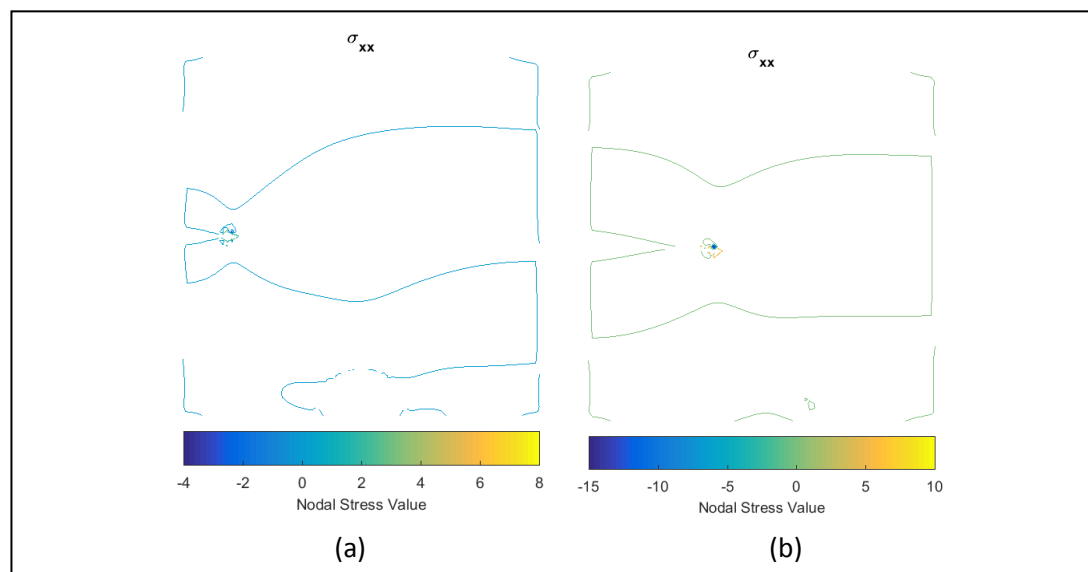
a. Percobaan Sobekan dengan Inklusi Lingkaran Variasi 1

Percobaan dilakukan dengan masukan seperti pada Tabel 4.5. Inklusi yang digunakan merupakan inklusi lingkaran. Variasi 1 merupakan variasi variabel modulus Young bernilai 10E9 dan rasio Poisson 0.3. Uji coba pertama ini diselesaikan dalam rata-rata waktu 24.51 detik untuk 30 kali iterasi.

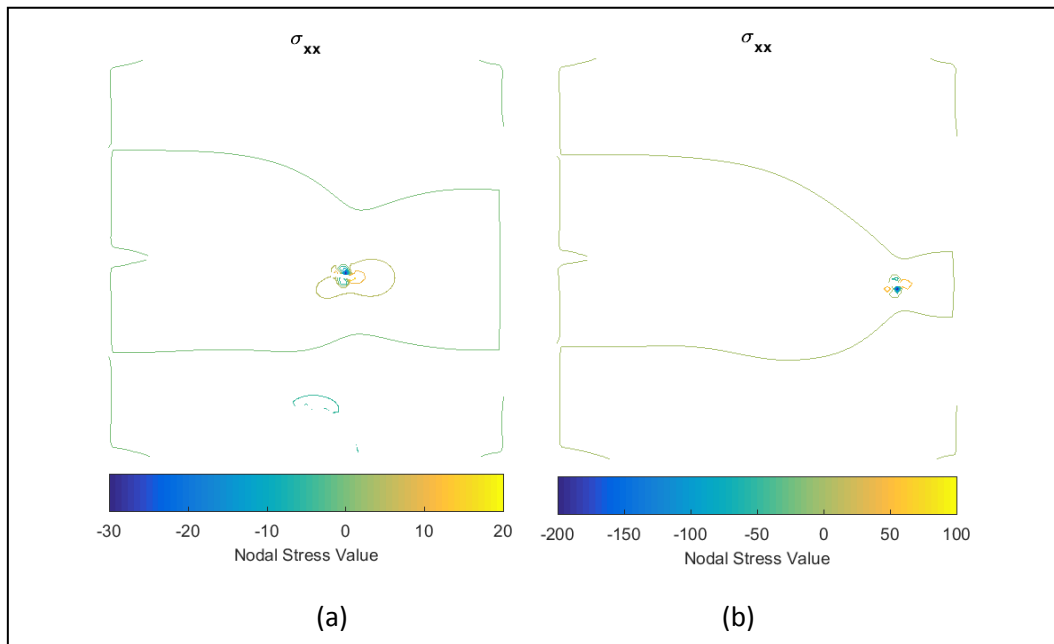
Tabel 4.5. Masukan untuk percobaan sobekan dengan inklusi lingkaran variasi 1

Variabel	Nilai
Height	4
Width	4
LElem	1/20;
DOMAIN	[Width/LElem,Height/LElem,LElem,LElem];
MAT	[10E9,0.3,10E8,0.3,2,0,Inf];
CRACK	[0 Height/2;0.5 Height/2]
INC	[2 0 0.5]
GROW	[30 0.1];
FORCE	[1 0 1]

Persebaran tegangan yang terjadi selama proses pembentukan sobekan ditunjukkan pada Gambar 4.10 dan Gambar 4.11.

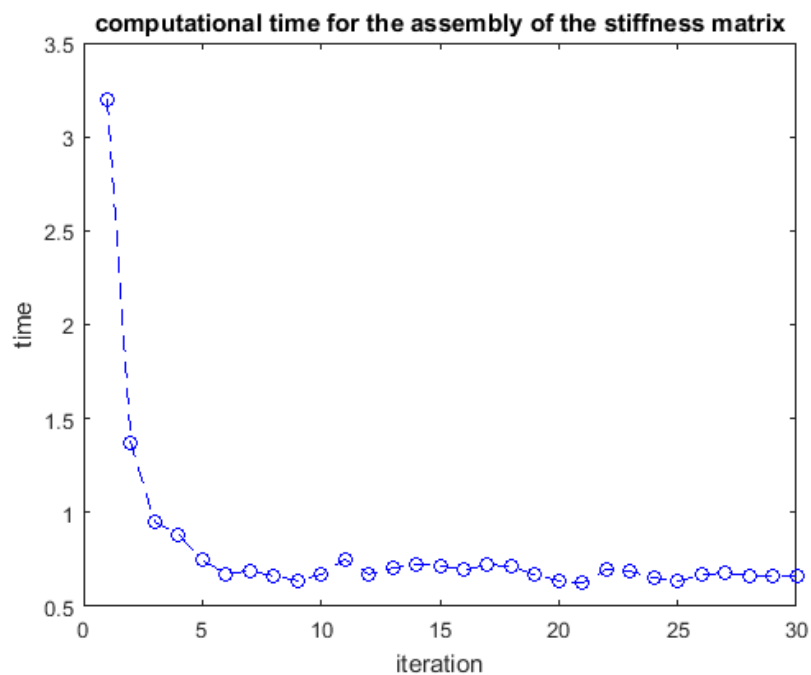


Gambar 4.10. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 1 (a) iterasi ke-1 (b) iterasi ke-10



Gambar 4.11. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 1 (a) iterasi ke-20 (b) iterasi ke-30

Sedangkan implementasi penggabungan atau *assembly* matrik kekakuan terhadap fungsi waktu yang diimplementasikan pada model untuk percobaan ini ditunjukkan pada Gambar 4.12.



Gambar 4.12. Grafik implementasi assembly matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi lingkaran variasi 1

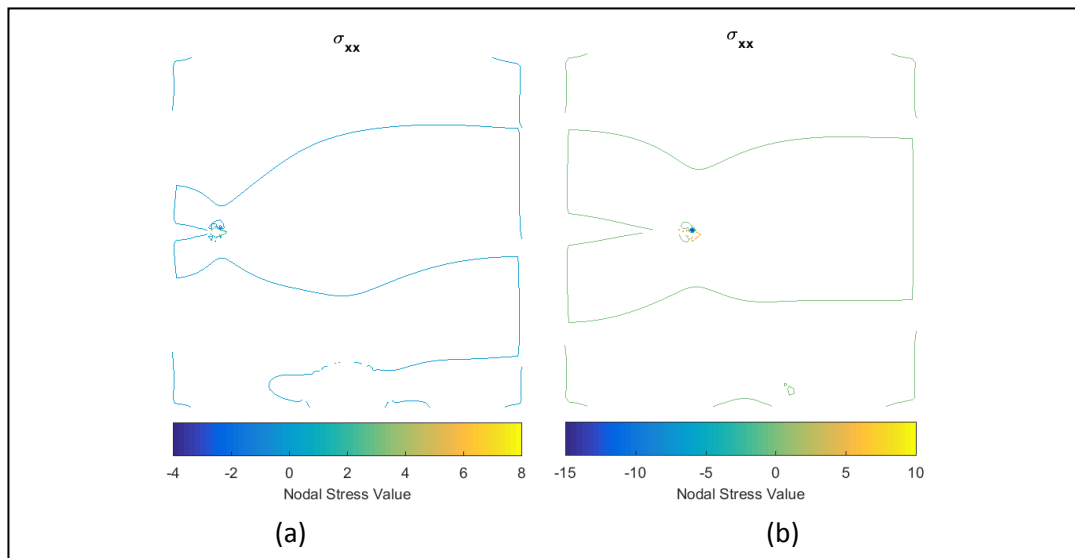
b. Percobaan Sobekan dengan Inklusi Lingkaran Variasi 2

Percobaan dilakukan dengan masukan seperti pada Tabel 4.6. Inklusi yang digunakan merupakan inklusi lingkaran. Variasi 2 merupakan variasi variabel modulus Young bernilai 5E9 dan rasio Poisson 0.3.. Uji coba pertama ini diselesaikan dalam rata-rata waktu 25.29 detik untuk 30 kali iterasi.

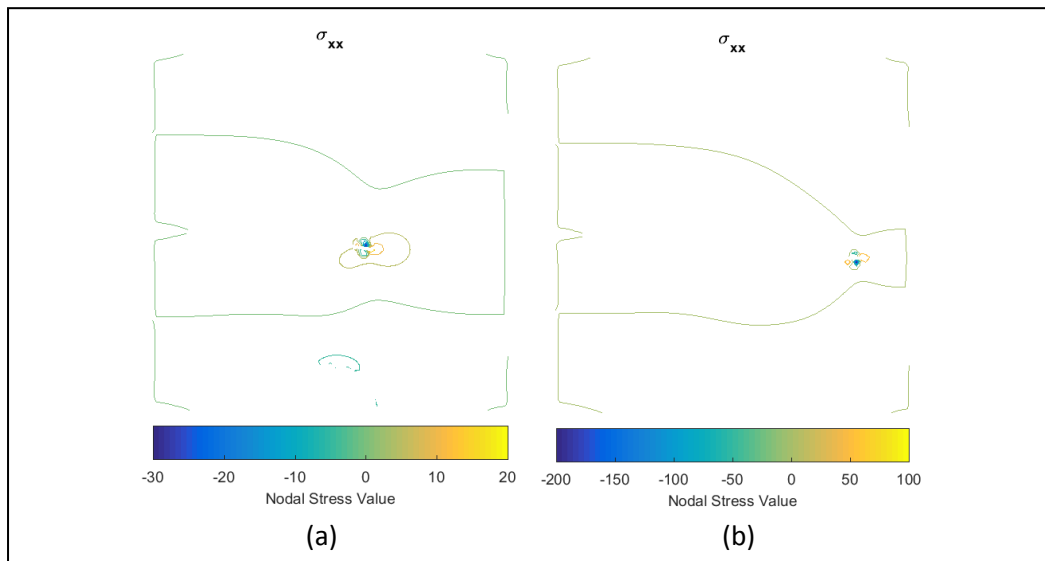
Tabel 4.6. Masukan untuk percobaan sobekan dengan inklusi lingkaran variasi 2

Variabel	Nilai
Height	4
Width	4
LElem	1/20;
DOMAIN	[Width/LElem,Height/LElem,LElem,LElem];
MAT	[5E9,0.3,5E8,0.3,2,0,Inf];
CRACK	[0 Height/2;0.5 Height/2]
INC	[2 0 0.5]
GROW	[30 0.1];
FORCE	[1 0 1]

Selain itu diilustrasikan pula persebaran tegangan yang terjadi selama proses pembentukan sobekan pada Gambar 4.13 dan Gambar 4.14.

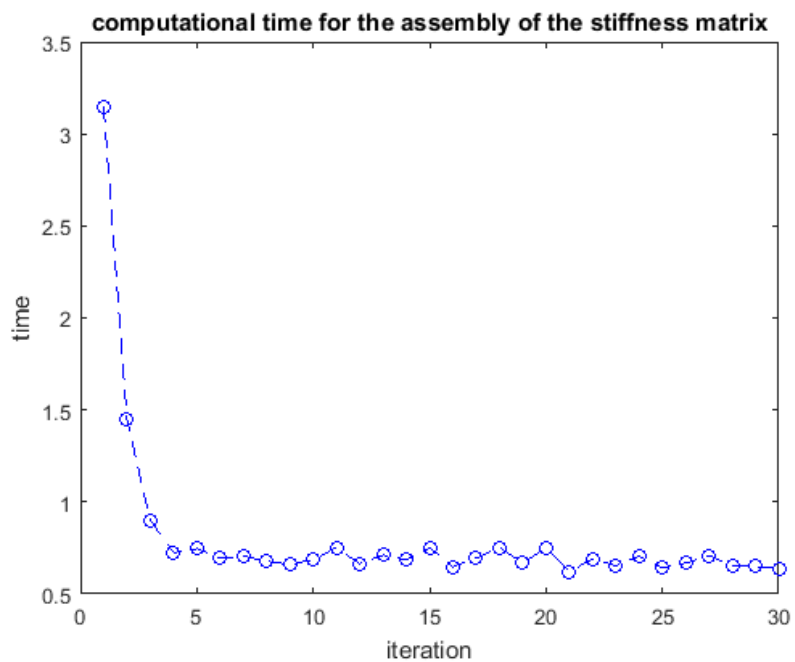


Gambar 4.13. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 2 (a) iterasi ke-1 (b) iterasi ke-10



Gambar 4.14. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 2 (a) iterasi ke-20 (b) iterasi ke-30

Sedangkan implementasi penggabungan atau *assembly* matrik kekakuan terhadap fungsi waktu yang diimplementasikan pada model untuk percobaan ini ditunjukkan pada Gambar 4.15.



Gambar 4.15. Grafik implementasi *assembly* matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi lingkaran variasi 2

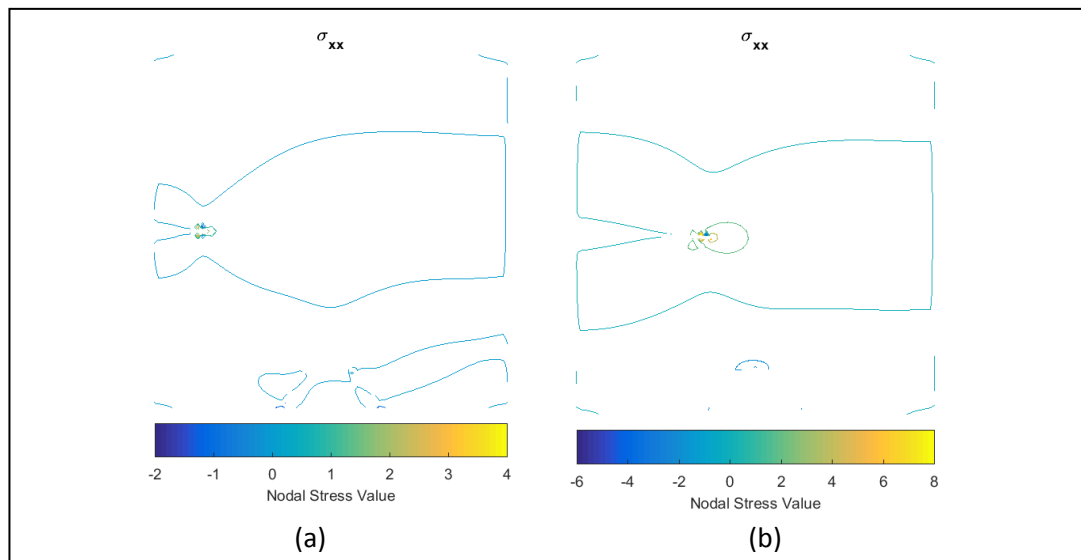
c. Percobaan Sobekan dengan Inklusi Lingkaran Variasi 3

Percobaan dilakukan dengan masukan seperti pada Tabel 4.7. Masukan untuk percobaan sobekan dengan inklusi lingkaran variasi 3. Inklusi yang digunakan merupakan inklusi lingkaran. Variasi 3 merupakan variasi variabel modulus Young bernilai 10E9 dan rasio Poisson 0.1. Uji coba pertama ini diselesaikan dalam rata-rata waktu 24.46 detik untuk 30 kali iterasi.

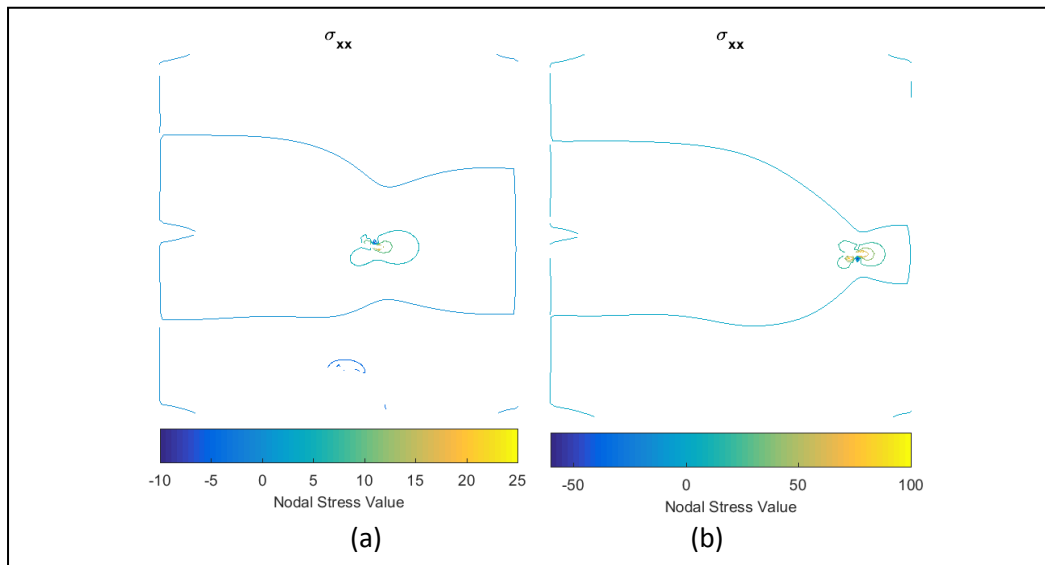
Tabel 4.7. Masukan untuk percobaan sobekan dengan inklusi lingkaran variasi 3

Variabel	Nilai
Height	4
Width	4
LElem	1/20;
DOMAIN	[Width/LElem,Height/LElem,LElem,LElem];
MAT	[10E9,0.1,10E8,0.1,2,0,Inf];
CRACK	[0 Height/2;0.5 Height/2]
INC	[2 0 0.5]
GROW	[30 0.1];
FORCE	[1 0 1]

Selain itu diilustrasikan pula persebaran tegangan yang terjadi selama proses pembentukan sobekan pada

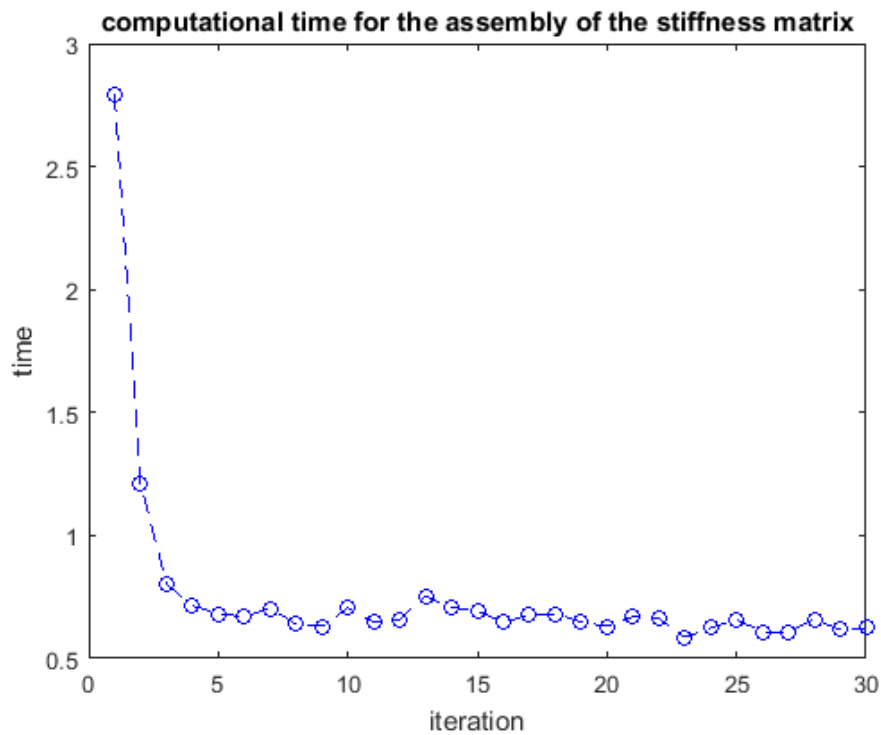


Gambar 4.16. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 3 (a) iterasi ke-1 (b) iterasi ke-10



Gambar 4.17. Persebaran Stress pada proses sobekan inklusi lingkaran variasi 3 (a) iterasi ke-20 (b) iterasi ke-30

Sedangkan implementasi penggabungan atau *assembly* matrik kekakuan terhadap fungsi waktu yang diimplementasikan pada model untuk percobaan ini ditunjukkan pada Gambar 4.18.



Gambar 4.18. Grafik implementasi *assembly* matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi lingkaran variasi 3

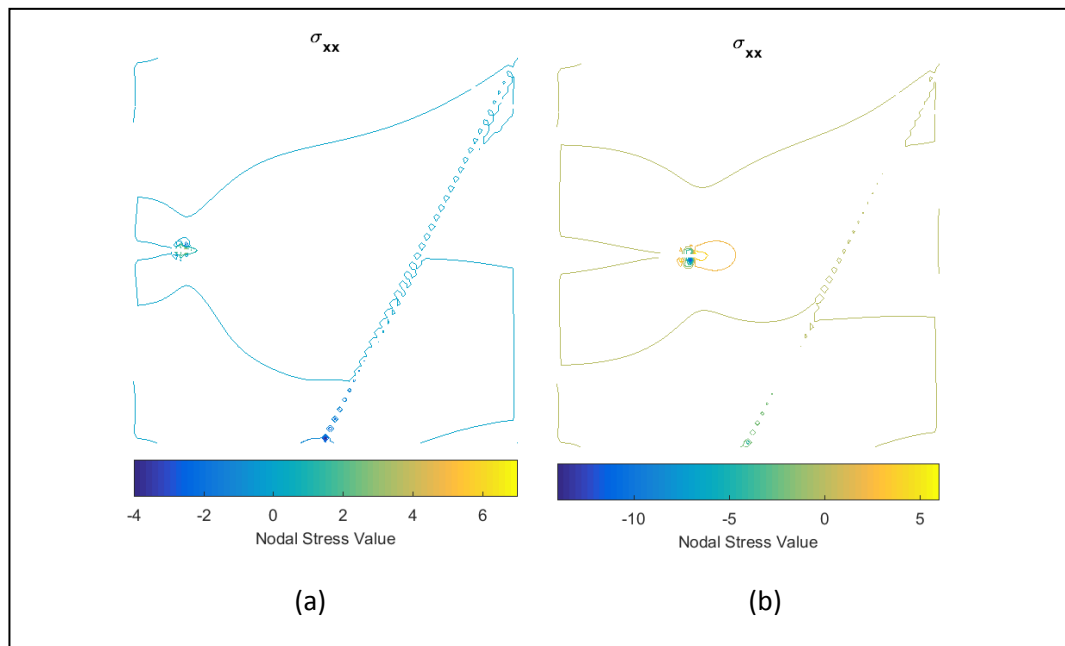
d. Percobaan Sobekan dengan Inklusi Linier Variasi 1

Percobaan dilakukan dengan masukan seperti pada Tabel 4.8. Inklusi yang digunakan merupakan inklusi linier. Variasi 1 merupakan variasi variabel modulus Young bernilai 10E9 dan rasio Poisson 0.3. Uji coba pertama ini diselesaikan dalam rata-rata waktu 33.68 detik untuk 30 kali iterasi.

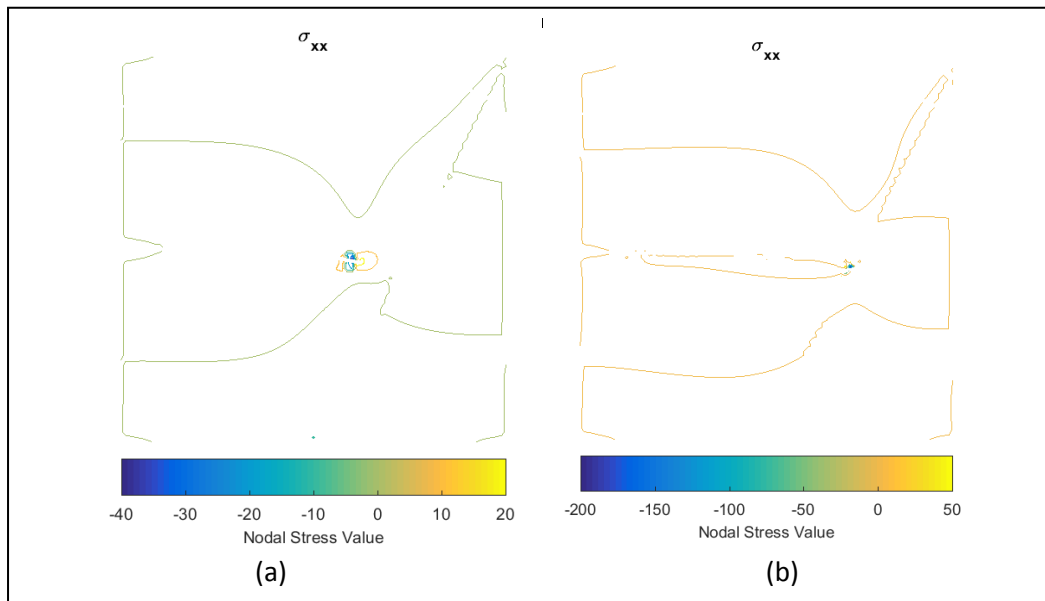
Tabel 4.8. Masukan untuk percobaan sobekan dengan inklusi linier variasi 1

Variabel	Nilai
Height	4
Width	4
LElem	1/20;
DOMAIN	[Width/LElem,Height/LElem,LElem,LElem];
MAT	[10E9,0.3,10E8,0.3,2,0,Inf];
CRACK	[0 Height/2;0.5 Height/2]
INC	[2 0 4 4]
GROW	[30 0.1];
FORCE	[1 0 1]

Selain itu diilustrasikan pula persebaran tegangan yang terjadi selama proses pembentukan sobekan pada Gambar 4.19 dan Gambar 4.20

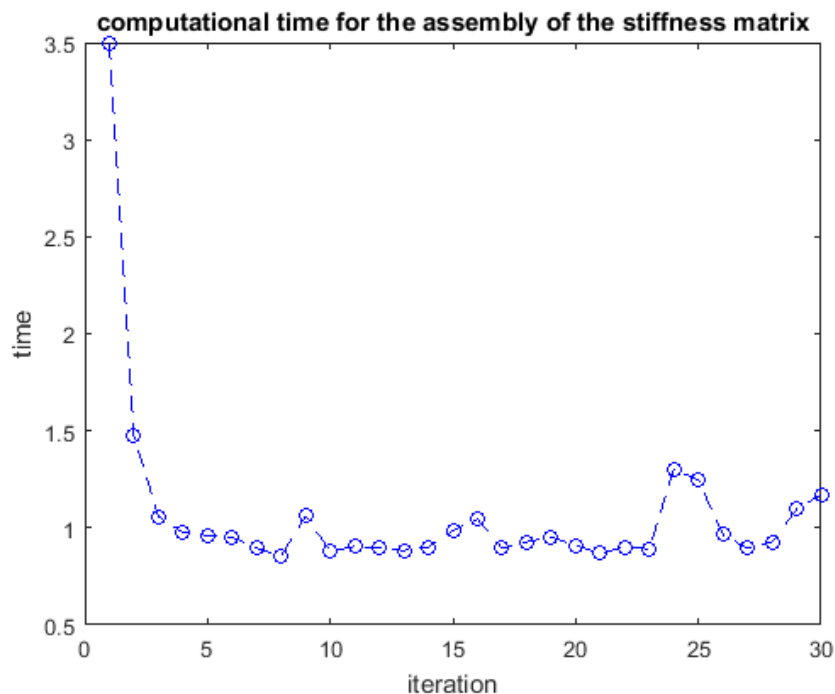


Gambar 4.19. Persebaran Stress pada proses sobekan inklusi linier variasi 1
(a) iterasi ke-1 (b) iterasi ke-10



Gambar 4.20. Persebaran Stress pada proses sobekan inklusi linier variasi 1
(a) iterasi ke-20 (b) iterasi ke-30

Sedangkan implementasi penggabungan atau *assembly* matrik kekakuan terhadap fungsi waktu yang diimplementasikan pada model untuk percobaan ini ditunjukkan pada Gambar 4.21.



Gambar 4.21. Grafik implementasi assembly matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi linier variasi 1

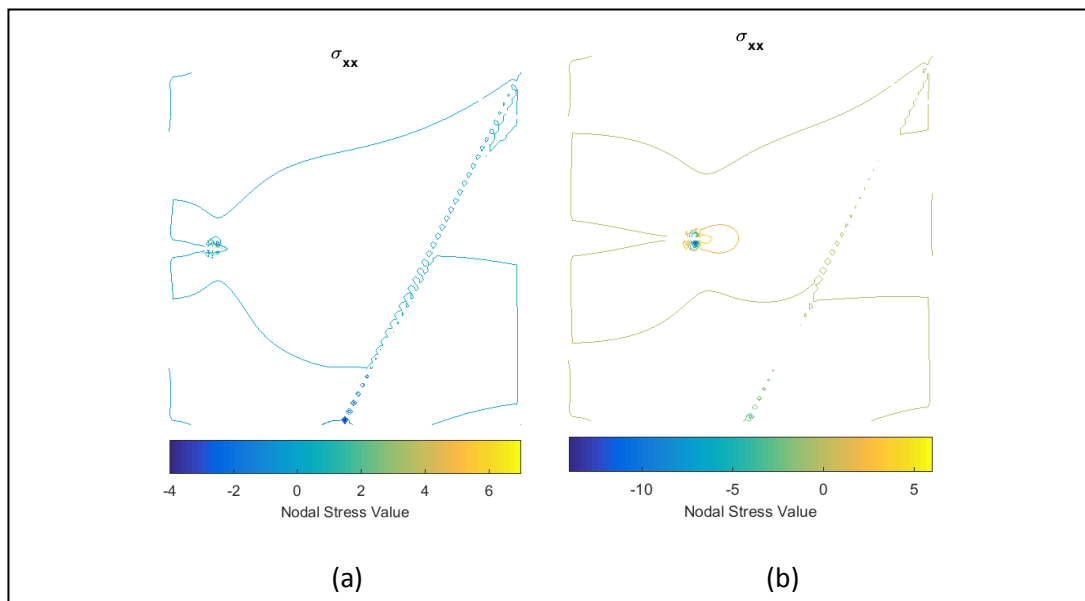
e. Percobaan Sobekan dengan Inklusi Linier Variasi 2

Percobaan dilakukan dengan masukan seperti pada Tabel 4.9. Inklusi yang digunakan merupakan inklusi linier. Variasi 1 merupakan variasi variabel modulus Young bernilai 10E9 dan rasio Poisson 0.3. Uji coba pertama ini diselesaikan dalam rata-rata waktu 36.44 detik untuk 30 kali iterasi.

Tabel 4.9. Masukan untuk percobaan sobekan dengan inklusi linier variasi 2

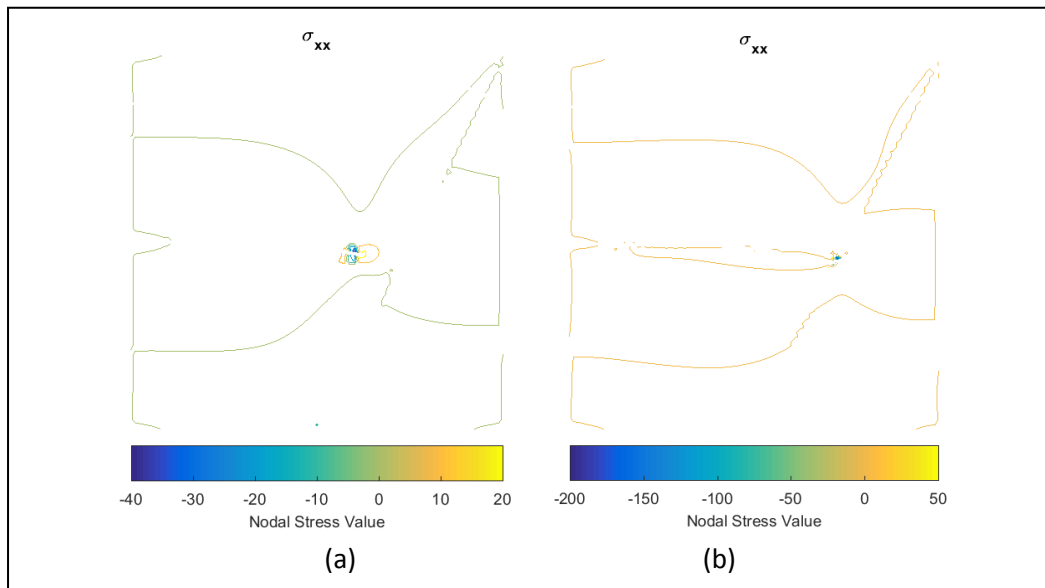
Variabel	Nilai
Height	4
Width	4
LElem	1/20;
DOMAIN	[Width/LElem,Height/LElem,LElem,LElem];
MAT	[5E9,0.3,5E8,0.3,2,0,Inf];
CRACK	[0 Height/2;0.5 Height/2]
INC	[2 0 4 4]
GROW	[30 0.1];
FORCE	[1 0 1]

Selain itu diilustrasikan pula persebaran tegangan yang terjadi selama proses pembentukan sobekan pada Gambar 4.22 dan Gambar 4.23.



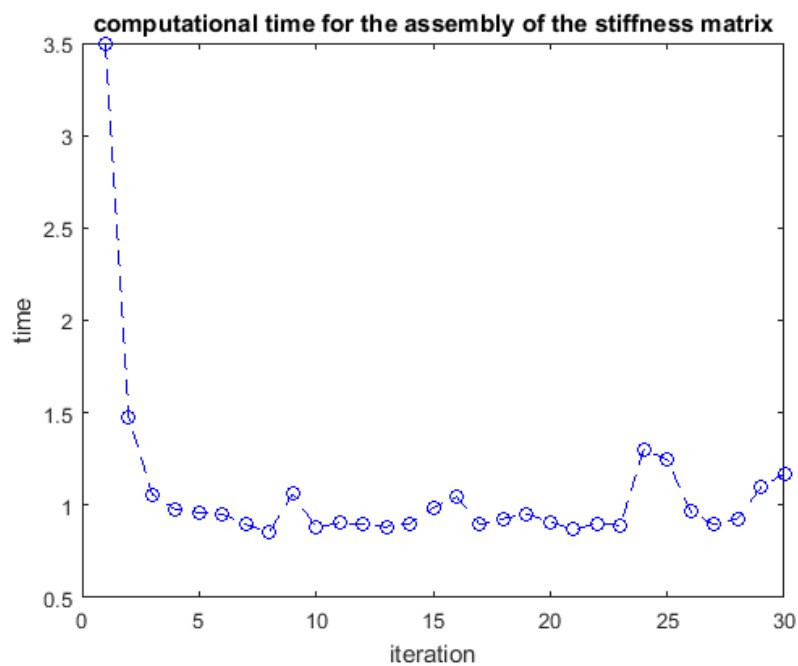
Gambar 4.22. Persebaran Stress pada proses sobekan inklusi linier variasi 2

(a) iterasi ke-1 (b) iterasi ke-10



Gambar 4.23. Persebaran Stress pada proses sobekan inklusi linier variasi 2
(a) iterasi ke-20 (b) iterasi ke-30

Sedangkan implementasi penggabungan atau *assembly* matrik kekakuan terhadap fungsi waktu yang diimplementasikan pada model untuk percobaan ini ditunjukkan pada Gambar 4.24.



Gambar 4.24. Grafik implementasi *assembly* matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi linier variasi 2

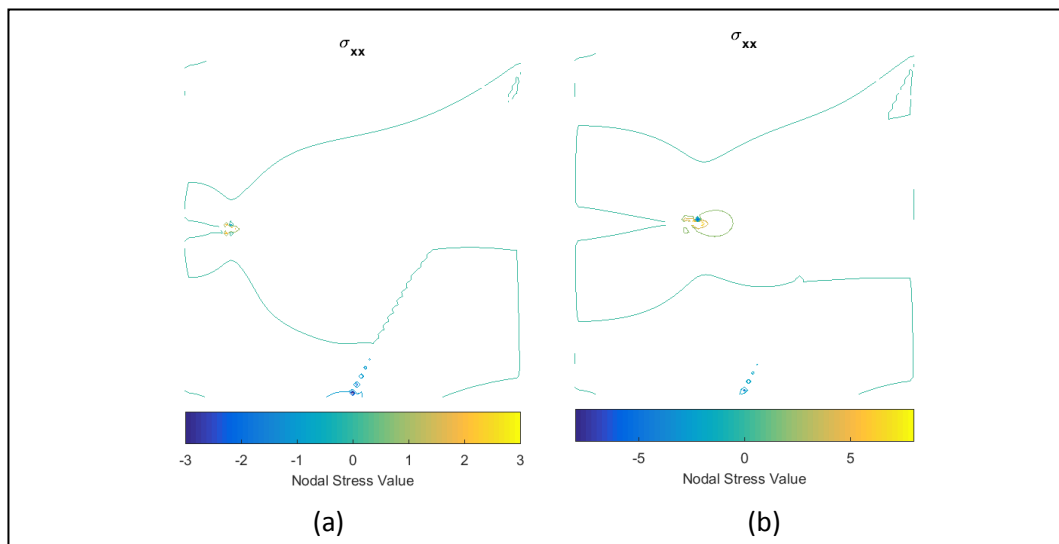
f. Percobaan Sobekan dengan Inklusi Linier Variasi 3

Percobaan dilakukan dengan masukan seperti pada Tabel 4.10. Inklusi yang digunakan merupakan inklusi linier. Variasi 3 merupakan variasi variabel modulus Young bernilai 10E9 dan rasio Poisson 0.1. Uji coba pertama ini diselesaikan dalam rata-rata waktu 39.05 detik untuk 30 kali iterasi.

Tabel 4.10. Masukan untuk percobaan sobekan dengan inklusi linier variasi 3

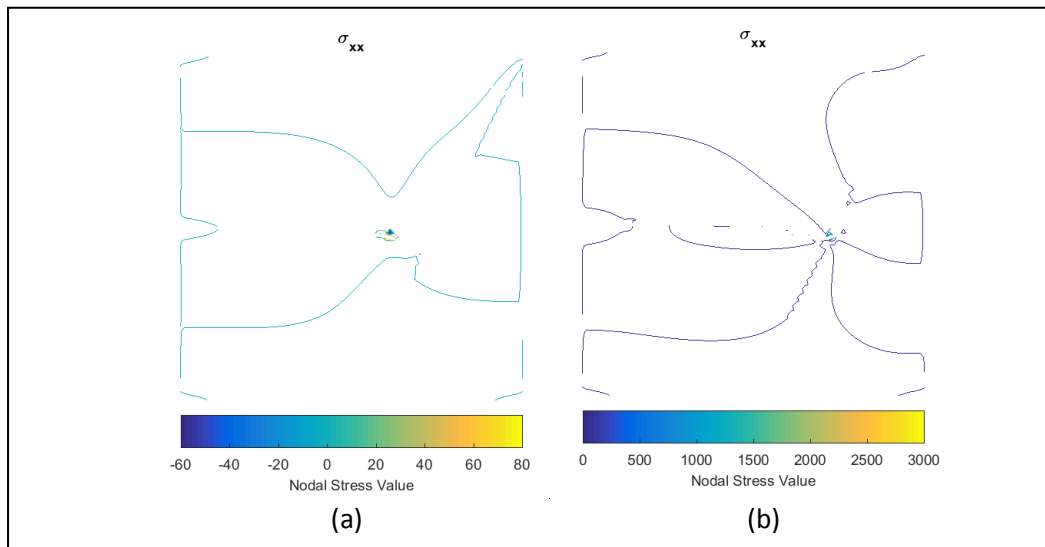
Variabel	Nilai
Height	4
Width	4
LElem	1/20;
DOMAIN	[Width/LElem,Height/LElem,LElem,LElem];
MAT	[10E9,0.1,10E8,0.1,2,0,Inf];
CRACK	[0 Height/2;0.5 Height/2]
INC	[2 0 4 4]
GROW	[30 0.1];
FORCE	[1 0 1]

Selain itu diilustrasikan pula persebaran tegangan yang terjadi selama proses pembentukan sobekan pada Gambar 4.25 dan Gambar 4.26.



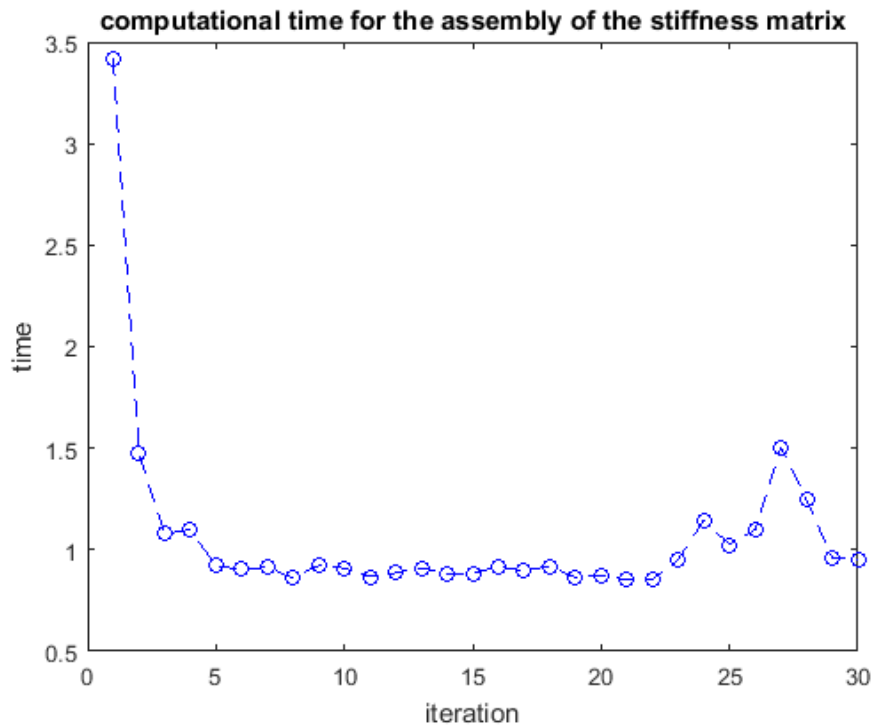
Gambar 4.25. Persebaran Stress pada proses sobekan inklusi linier variasi 3

(a) iterasi ke-1 (b) iterasi ke-10



**Gambar 4.26. Persebaran Stress pada proses sobekan inklusi linier variasi 3
(a) iterasi ke-20 (b) iterasi ke-30**

Sedangkan implementasi penggabungan atau *assembly* matrik kekakuan terhadap fungsi waktu yang diimplementasikan pada model untuk percobaan ini ditunjukkan pada



Gambar 4.27. Grafik implementasi *assembly* matrik kekakuan terhadap fungsi waktu percobaan dengan inklusi linier variasi 3

4.3. Analisis Hasil Uji Coba

Uji coba utama dilakukan dengan dua skenario yaitu tanpa inklusi dan dengan inklusi. Dari kedua skenario utama tersebut terdapat beberapa subskenario yang dipengaruhi oleh variabel modulus Young dan rasio Poisson yang ada pada masukan pengguna. Pada skenario dengan inklusi, terdapat dua skenario yang ditentukan oleh jenis inklusi yang diterapkan—lingkaran dan linier.

Hasil uji coba ditunjukkan berupa kontur nilai *stress* nodal. Nilai *stress* nodal merupakan nilai *stress* yang dipengaruhi oleh nilai Gauss pada sebuah titik. Pada gambar kontur ini ditunjukkan bahwa setiap titik sobekan bergantung pada matrik kekakuan yang dikomputasi pada setiap fungsi waktu. Gambar kontur tersebut juga menunjukkan bahwa persebaran matrik kekakuan pada setiap iterasi tetap berkaitan dengan persebaran matrik pada iterasi sebelumnya.

Sedangkan grafik implementasi *assembly* matrik kekakuan terhadap fungsi waktu menunjukkan sifat konvergensi model terhadap penghitungan matrik kekakuan pada setiap iterasi. Hasil uji coba menunjukkan bahwa model mulai konvergen rata-rata pada iterasi ke 5. Hasil uji coba juga menunjukkan bahwa tingkat konvergensi akan berlaku pada model baik dengan inklusi maupun tanpa inklusi serta dengan berbagai variasi variabel elastisitas.

Dari hasil uji coba, didapatkan rata-rata waktu komputasi yang dibutuhkan untuk membentuk sobekan pada objek. Waktu komputasi ini dapat dilihat secara keseluruhan pada Tabel 4.11 dan Tabel 4.12. Dapat dilihat pada Tabel 4.11 dan Tabel 4.12 bahwa pembentukan sobekan pada material tanpa inklusi lebih cepat dari pada dengan inklusi. Sementara waktu yang dibutuhkan untuk inklusi lingkaran lebih cepat daripada inklusi linier.

Ilustrasi elemen-elemen pengayaan pada model dan ilustrasi *mesh* yang sudah dideformasi ditunjukkan pada Lampiran C.

Tabel 4.11. Waktu komputasi uji coba (1)

Percobaan	Waktu Komputasi (detik)
4.2.1. a	23.67
4.2.1. b	21.34
4.2.1. c	24.49
4.2.2. a	24.51

Tabel 4.12. Waktu komputasi uji coba (2)

Percobaan	Waktu Komputasi (detik)
4.2.2. b	25.29
4.2.2. c	24.46
4.2.2. e	36.44
4.2.2. f	39.05

BAB 5

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang didapat dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang telah dikemukakan. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian selanjutnya.

5.1. Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap sistem yang dibangun untuk proses pembentukan model sobekan pada penelitian ini, dapat diambil kesimpulan sebagai berikut:

1. Metode modifikasi Finite Element dengan elemen pengayaan dapat merepresentasikan model penyobekan pada bidang tipis seperti kertas.
2. Metode yang diimplementasikan menghasilkan model yang konvergen terhadap variasi inklusi maupun variasi variabel.
3. Waktu komputasi yang dibutuhkan untuk membentuk visualisasi sobekan dipengaruhi oleh ada tidaknya inklusi dan jenis inklusi yang diterapkan. Dari hasil uji coba, waktu komputasi tercepat dihasilkan dari model yang tidak mengaplikasikan titik inklusi. Sementara waktu komputasi terlama dihasilkan dari model yang mengaplikasikan titik inklusi linier.

5.2. Saran

Saran yang diberikan untuk penelitian pembangunan model sobekan pada kertas antara lain:

1. Perbaikan model dengan metode modifikasi Finite Element memperhitungkan komponen lain pada domain sehingga dapat menghasilkan detail pada bidang visualisasi.
2. Pengaplikasian metode sobekan pada material lain yang lebih beragam.
3. Implementasi visualisasi agar hasil sobekan dapat dinyatakan secara realistis.

[halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Fries, T.-P. et al., 2011. Hanging nodes and XFEM. *International Journal for Numerical Method in Engineering*, 86(4-5), pp. 404-430.
- Melenk, J. M. & Babuska, I., 1996. The partition of unity finite element method: Basic theory and application. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4), pp. 289-314.
- Moes, N., Cloirec, M., Cartraud, M. & Remacle, J., 2003. A Computational Approach to Handle Complex Microstructure Geometries. *Computer Methods in Applied Mechanics and Engineering*, Volume 192, pp. 3163-3177.
- Moes, N., Dolbow, J. & Belytschko, T., 1999. A Finite Element Method for crack growth without remeshing. *International Journal for Numerical Method in Engineering*, Volume 46, pp. 131-150.
- Narain, R., Pfaff, T. & O'Brien, J. F., 2013. Folding and Crumpling Adaptive Sheets. *ACM Transaction on Graphics*, 32(4).
- O'Brien, J. F., 2000. *Graphical Modeling and Animation of Fracture*, Georgia: Georgia Institute of Technology.
- O'Brien, J. F. & Hodgins, J. K., 1999. *Graphical Modeling and Animation of Brittle Fracture*. Los Angeles, California, ACM SIGGRAPH.
- Pais, M. J., 2011. *Variable amplitude fatigue analysis using surrogate models and exact xfem reanalysis*. Florida: UNIVERSITY OF FLORIDA.
- Pais, M. J., Kim, N.-H. & Davis, T., 2010. *Reanalysis of the Extended Finite Element Method for Crack Initiation Propagation*. Florida, AIAA SDM Student Symposium.
- Pfaff, T., Narain, R., de Joya, J. M. & O'Brien, J. F., 2014. Adaptive Tearing and Cracking of Thin Sheets. *ACM Transactions On Graphics*, July. Volume 33.

Sukumar, N., Chopp, D., Moes, N. & Belytschko, T., 2001. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer methods in applied mechanic and engineering*, pp. 6183-6200.

LAMPIRAN A

Nilai modulus Young berdasarkan Materials Data Book dari Cambridge University Engineering Department (Cambridge University Engineering Departement, 2003).

II.3 YOUNG'S MODULUS, E

		E (GPa)
Metals	Ferrous	Cast Irons 165 - 180 High Carbon Steels 200 - 215 Medium Carbon Steels 200 - 216 Low Carbon Steels 200 - 215 Low Alloy Steels 201 - 217 Stainless Steels 188 - 210 Aluminium Alloys 68 - 82 Copper Alloys 112 - 148 Lead Alloys 12.5 - 15 Magnesium Alloys 42 - 47 Nickel Alloys 190 - 220 Titanium Alloys 90 - 120 Zinc Alloys 68 - 95
	Non-ferrous	Borosilicate Glass 61 - 64 Glass Ceramic 64 - 110 Silica Glass 68 - 74 Soda-Lime Glass 68 - 72 Brick 10 - 50 Concrete, typical 25 - 38 Stone 6.9 - 21 Alumina 215 - 413 Aluminium Nitride 302 - 348 Boron Carbide 400 - 472 Silicon 140 - 155 Silicon Carbide 300 - 460 Silicon Nitride 280 - 310 Tungsten Carbide 600 - 720
Composites	Metal Polymer	Aluminium/Silicon Carbide 81 - 100 CFRP 69 - 150 GFRP 15 - 28
	Natural	Bamboo 15 - 20 Cork 0.013 - 0.05 Leather 0.1 - 0.5 Wood, typical (Longitudinal) 6 - 20 Wood, typical (Transverse) 0.5 - 3

		E (GPa)
Polymers ¹	Elastomer	Butyl Rubber 0.001 - 0.002 EVA 0.01 - 0.04 Isoprene (IR) 0.0014 - 0.004 Natural Rubber (NR) 0.0015 - 0.0025 Neoprene (CR) 0.0007 - 0.002 Polyurethane Elastomers (aPU) 0.002 - 0.003 Silicone Elastomers 0.005 - 0.02 ABS 1.1 - 2.9 Cellulose Polymers (CA) 1.6 - 2 Ionomer (I) 0.2 - 0.424 Nylons (PA) 2.62 - 3.2 Polycarbonate (PC) 2 - 2.44 PEEK 3.5 - 4.2 PET 0.621 - 0.896 Polyethylene (PE) 2.76 - 4.14 Acrylic (PMMA) 2.24 - 3.8 Acetal (POM) 2.5 - 5 Polypropylene (PP) 0.866 - 1.55 Polystyrene (PS) 2.28 - 3.34 Polyurethane Thermoplastics (tpPU) 1.31 - 2.07 PVC 2.14 - 4.14 Teflon (PTFE) 0.4 - 0.552 Epoxies 2.35 - 3.075 Phenolics 2.76 - 4.83 Polyester 2.07 - 4.41
	Thermoset	Flexible Polymer Foam (VLD) 0.0003 - 0.001 Flexible Polymer Foam (LD) 0.001 - 0.003 Flexible Polymer Foam (MD) 0.004 - 0.012 Rigid Polymer Foam (LD) 0.023 - 0.08 Rigid Polymer Foam (MD) 0.08 - 0.2 Rigid Polymer Foam (HD) 0.2 - 0.48

1 For full names and acronyms of polymers – see Section V
(Data courtesy of Granta Design Ltd)

Gambar A.1. Referensi nilai konstan modulus Young

[Halaman ini sengaja dikosongkan]

LAMPIRAN B

Nilai konstanta rasio Poisson berdasarkan Mechanics of Material (Gere & Timoshenko, 1997)

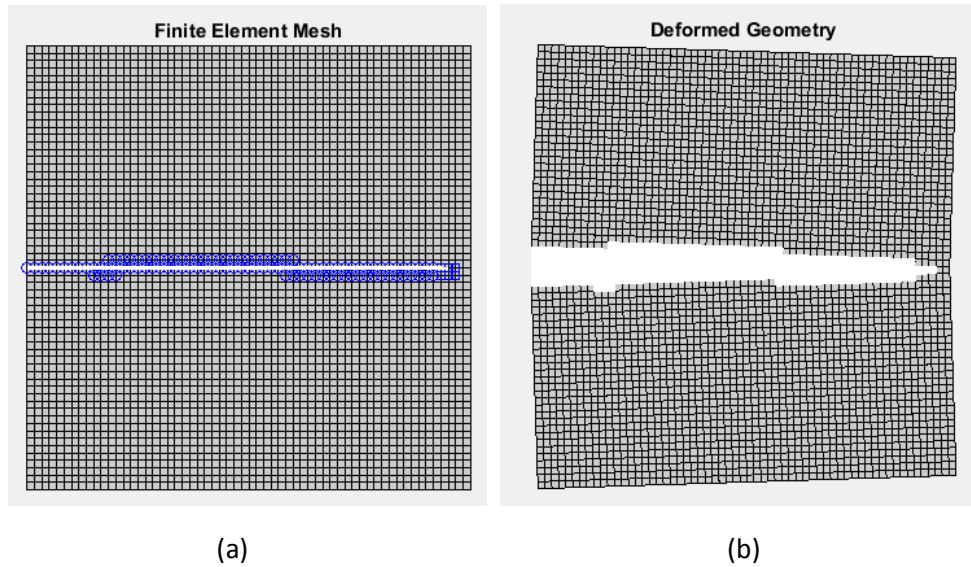
MODULI OF ELASTICITY AND POISSON'S RATIO					
Material	Modulus of elasticity E		Shear modulus of elasticity G		Poisson's ratio ν
	ksi	GPa	ksi	GPa	
Aluminum alloys	10,000- 11,400	70-79	3,800-4,300	26-30	0.33
2014- T6	10,600	73	4,000	28	0.33
6061- T6	10,000	70	3,800	26	0.33
7075- T6	10,400	72	3,900	27	0.33
Brass	14,000-16,000	96-110	5,200-6,000	36-41	0.34
Bronze	14,000-17,000	96-120	5,200-6,300	36-44	0.34
Cast iron	12,000-25,000	83-170	4,600-10,000	32-69	0.2-0.3
Concrete (compression)	2,500-4,500	17-31			0.1-0.2
Copper and Copper alloys	16,000-18,000	110-120	5,800-6,800	40-47	0.33-0.36
Glass	7,000-12,000	48-83	2,700-5,100	19-35	0.17-0.27
Magnesium alloys	6,000-6,500	41-45	2,200-2,400	15-17	0.35
Nickel	30,000	210	11,400	80	0.31
Plastics					
Nylon	300-500	2.1-3.4			0.4
Polyethylene	100-200	0.7-1.4			0.4
Rock (Compression)					
Granite, marble, quartz	6,000-14,000	40-100			0.2-0.3
Limestone, sandstone	3,000-10,000	20-70			0.2-0.3
Rubber	0.1-0.6	0.0007-0.004	0.03-0.20	0.0002-0.001	0.45-0.50
Steel	28,000-30,000	190-210	10,800-11,800	75-80	0.27-0.30
Titanium alloys	15,000-17,000	100-120	5,600-6,400	39-44	0.33
Tungsten	50,000-55,000	340-380	21,000-23,000	140-160	0.2
Wood (bending)					
Douglas fir	1,600-1,900	11-13			
Oak	1,600-1,800	11-12			
Southern pine	1,600-2,000	11-14			

Gambar B.1 Konstanta elastisitas berdasarkan Mechanic of Materials

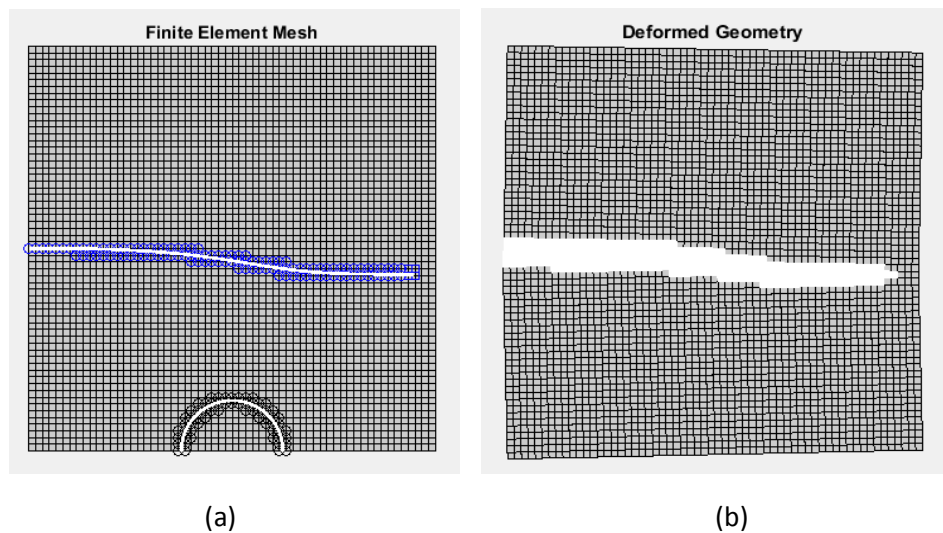
[Halaman ini sengaja dikosongkan]

LAMPIRAN C

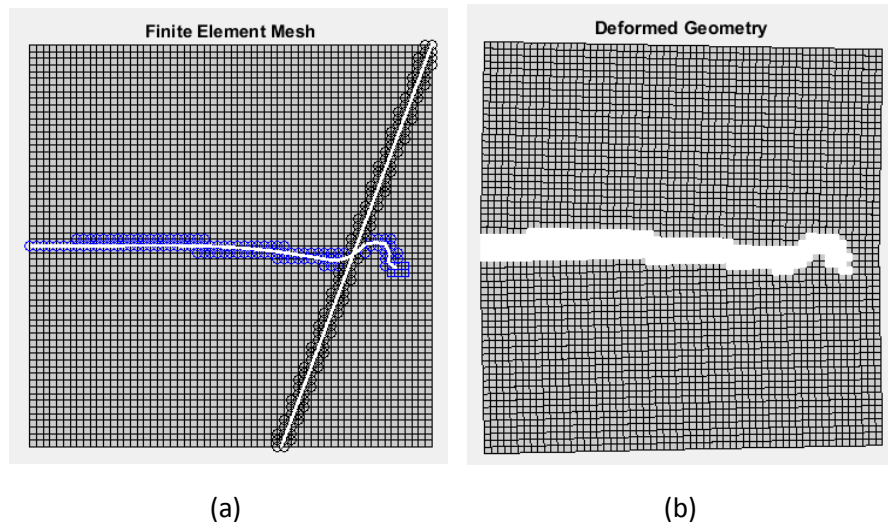
Ilustrasi garis sobekan dan *mesh* dalam bentuk deformasi pada model. Ilustrasi untuk model sobekan tanpa inklusi ada pada Gambar C.1. Ilustrasi untuk model sobekan dengan inklusi lingkaran ada pada Gambar C.2. Sedangkan untuk ilustrasi model sobekan dengan inklusi linier ditunjukkan pada Gambar C.3.



Gambar C.1. Ilustrasi model sobekan tanpa inklusi (a) Finite Element Mesh dan elemen pengayaan (b) Mesh deformasi



Gambar C.2. Ilustrasi model sobekan dengan inklusi lingkaran (a) Finite Element Mesh dan elemen pengayaan (b) Mesh deformasi



Gambar C.3. Ilustrasi model sobekan dengan inklusi linier (a) Finite Element Mesh dan elemen pengayaan (b) Mesh deformasi

BIODATA PENULIS



Astris Dyah Perwita, lahir pada 25 September 1991 merupakan anak kedua dari dua bersaudara. Penulis memulai pendidikan dasar pada tahun 1998 di MIN Malang I dan melanjutkan pendidikan menengah pertama pada tahun 2004 di SMPN 3 Malang. Setelah itu melanjutkan pendidikan menengah atas pada tahun 2007 di SMAN 3 Malang. Penulis melanjutkan pendidikan jenjang perguruan tinggi di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya mulai tahun 2010 dan mendapatkan gelar sarjana pada tahun 2014. Pada tahun 2014 penulis melanjutkan pendidikan tingkat magister di institusi yang sama.

Penulis memiliki ketertarikan pada bidang komputasi dan algoritma serta komputer grafis. Penulis dapat dihubungi melalui email astrisd@gmail.com